UNIVERSITÀ DEGLI STUDI DI GENOVA

PhD in Digital Humanities

Cycle XXXII

A framework for structuring prerequisite relations between concepts in educational textbooks

Candidate: Samuele Passalacqua

Supervisors: Prof. Ilaria Torre, Dr. Frosina Koceva

Abstract

In our age we are experiencing an increasing availability of digital educational resources and self-regulated learning. In this scenario, the development of automatic strategies for organizing the knowledge embodied in educational resources has a tremendous potential for building personalized learning paths and applications such as intelligent textbooks and recommender systems of learning materials. To this aim, a straightforward approach consists in enriching the educational materials with a concept graph, i.a. a knowledge structure where key concepts of the subject matter are represented as nodes and prerequisite dependencies among such concepts are also explicitly represented. This thesis focuses therefore on prerequisite relations in textbooks and it has two main research goals. The first goal is to define a methodology for systematically annotating prerequisite relations in textbooks, which is functional for analysing the prerequisite phenomenon and for evaluating and training automatic methods of extraction. The second goal concerns the automatic extraction of prerequisite relations from textbooks. These two research goals will guide towards the design of PRET, i.e. a comprehensive framework for supporting researchers involved in this research issue. The framework described in the present thesis allows indeed researchers to conduct the following tasks: 1) manual annotation of educational texts, in order to create datasets to be used for machine learning algorithms or for evaluation as gold standards; 2) annotation analysis, for investigating inter-annotator agreement, graph metrics and in-context linguistic features; 3) data visualization, for visually exploring datasets and gaining insights of the problem that may lead to improve algorithms; 4) automatic extraction of prerequisite relations. As for the automatic extraction, we developed a method that is based on burst analysis of concepts in the textbook and we used the gold dataset with PR annotation for its evaluation, comparing the method with other metrics for PR extraction.

ACKNOWLEDGEMENTS

I would like to thank first of all my advisor Ilaria Torre, my co-advisor Frosina Koceva and prof. Giovanni Adorni for guiding me throughout my PhD and helping me to do my research.

I also express my gratitude to:

- Professor Peter Brusilovsky, who hosted me and guided me during my visiting period in School of Computing and Information, University of Pittsburgh.
- professors Reva Freedman and Noboru Matsuda, for having read and reviewed my thesis.

These years were for me rich of opportunities to get involved and spend time with different people in several contexts, and at least a mention goes to: professors and colleagues of the Digital Humanities PhD program at the university of Genoa/Turin, involved in both distant and near topics of research (among colleagues of the latter group, Chiara Alzetta, who shares the topic of my research); AI lab at the University of Genoa, DIBRIS department, with its researchers and PhD students; PAWS lab team members at the University of Pittsburgh (Jordan, Kamil, Khushboo, Zak, Ben, Hung...you really helped me to feel like home in Pitts and in the lab).

Finally, a very special thank goes to: my parents and my family for keeping on supporting me throughout all these years; Valentina, for encouraging me to spend more effort on the thesis and for bearing, during this period, almost every word of my "pipponi"; my friends at losfuso beershop in Turin; my work, colleagues and students at school during the months while I was working on the thesis (you gave me extra work but at the end it helped me to keep my mind in the external world during the thesis isolation).

TABLE OF CONTENTS

List of Tables

List of Figures

Figure **Page**

CHAPTER 1

INTRODUCTION

There is a wrong but interesting etymological explanation of the word *magister* (i.e. the latin word for “teacher”) which can be heard sometimes among teachers with a background in the Humanities. Curiously, this explanation is not only historically inaccurate but it is also associated to a funny story, albeit maybe this is funny only from the point of view of a humanist. The joke goes like this: listen to the latin sentence *Magis ter meus asinus est*, and you will understand it either as “My donkey eats three times more” or “My teacher is a donkey”. The mistake is triggered because *magister* (literally, "the greatest") and *magis ter* (“three times more”) seems to have the same meaning. A teacher, hence, seems to be three times more.

It is generally accepted that a person should master many skills if he or she wants to be a teacher. More than anybody else, teachers must indeed satisfy at least three fundamental requirements:

1. know their domain by heart;
2. know how to teach their domain;
3. know their students.

In some countries, people who want to become teachers must follow a long training and apprenticeship[1]. At least during the first steps, the training program has lots in

[1]Italy, for instance, is among such countries, and here teacher training follow a path that is very similar to the one described in the text.

common with a general curriculum. Hence, one who wants to become, let say, a maths teacher must first earn a degree in Mathematics, aimed at gaining basic as well as advanced knowledge in the field. During this phase almost no exams are dedicated to general pedagogy or to specific maths teaching methodologies. This process of building up domain knowledge can take years, but it is worthwhile. Even if a middle school maths teacher, during her daily work, only needs basic maths, a solid preparation across the full spectrum of her domain gives her enough authority to establish unusual links among concepts and skills, satisfying the curiosity of her pupils and avoiding restricting the learning process to a mere transmission of a limited range of basic notions. Subsequently, teachers in training undertake a specific program to become professionals, at the end of which they will possess pedagogical strategies (common to every subject teaching) and specific teaching methodologies of their own subject. Finally, they will gain experience in class regarding how to behave with students, in particular how to understand their needs, evaluate past progress, recognizing potential and help them in formulating and achieving their goals. This involves sharping some practical tools (e.g. assessment techniques) but also enforcing human qualities and social skills.

We can indulge the wrong etymology mentioned at the beginning of the chapter and add even a fourth competence, that is implicitly stated between the lines of competences 1-3 listed above: teachers must be capable of transforming their broad knowledge (competence 1) according to whom they are addressing (competence 3) using specific pedagogical strategies (competence 2). In other words, teachers must know how to transform and adapt their knowledge, with respect to each one of their students. Once they reach a high level of knowledge, experts often forget how difficult it was to learn a topic or a concept at the beginning of their training. This can make them unable to transfer their knowledge to students using adequate basic strategies, such as text simplification, adequate textbook selection, or recalling of previously explained concepts. Overlooking the importance of some prerequisite concepts before teaching a new topic is often enough to compromise a good understanding. No matter how cultured a teacher might be, he must carefully not fall into this knowledge paradox (i.e. the more one knows, the less he knows) and should instead always strive to transform his domain knowledge in the most adequate way for a particular student, taking into account what the student has to know in order to follow explanations or learning materials. As an example, think about a group of high school students asking their Computer Science teacher to give a lesson about Machine Learning, because they heard about it, it is a cool topic and they think it could be useful for them or just because they want to figure out what it is.

Denying explanations just because it represents a too advanced topic for them would be a pity; on the other hand, giving a detailed description without any adaptation and simplification of learning materials (thus regardless their background knowledge) would be only a solipsistic display of erudition: the students will be literally dazzled by the teacher's erudition, but most of this information will probably fail to reach their long term retention. The only take home message for them will be something like "well, it's a complex stuff".

1.1 Context of the research

Education has been for ages a teachers' prerogative. The entrance of ICT (Information and Communication Technologies) and Artificial Intelligence (AI) into the learning process throws new challenges, forcing teachers and researchers to find answers to hard questions. This thesis focuses on a topic of crucial importance for intelligent e-learning systems. Using a simple yet clear definition, we can describe such system as online learning environments that are capable of delivering the right content, to the right person, at the right time [201]. Intelligent learning systems (ILS) are far from being a brand new topic in research, as they have been around at least since the 1970s [203]. With the development of Computer Aided Instruction (CAI) first, and Intelligent Tutoring Systems (ITS) and Adaptive Hypermedia (AH) later, researchers in Artificial Intelligence in education have been pursuing the goal of creating automatic applications capable of assisting students by simulating the behaviour of a good human teacher. The most radical attempts pushed even further this idea, expressing the ultimate desire to build an automatic program that would act exactly like a good human teacher. In the field of intelligent learning systems, both in their strong (i.e. radical) or weak interpretation, adaptivity and personalisation are two frequently sought aspirations. In order to accomplish these two qualities, an online education environment needs to define and implement complex architectures, with multiple models, each one addressing and modelling a particular aspect of the triadic learner-teacher-subject interaction. Traditional knowledge-based adaptive intelligent tutoring systems can be described in terms of [201]:

1. a knowledge model (also called domain model, expert model [163] or content model [201]), which contains a formal representation of the knowledge and skills associated to a domain and to a specific set of learning materials belonging to that

domain (this model also includes the structural interdependence relations between such elements).

2. a learner model, which consists of the student's cognitive, motivational and other psychological states in a given moment and in relation to the domain model. This information (which also aggregates demographic information) can be estimated during the learner's activity (for example when solving items in assessment tests) or more generally during interactions with the system.

3. an instructional (or tutoring or pedagogical) model, which contains a set of pedagogical strategies to be used in order to present learning contents and monitor the learner through the learning process.

4. an engine which integrates information obtained from the previous models and creates personalised and adaptive presentations of the learning content.

Not surprisingly, these components match the fundamental competences of a good human teacher that we listed above. Thanks to a knowledge model, the system knows the domain; thanks to a learner's model, it is able to figure out what kind of student is using the system; thanks to an instructional model, it can draw the proper pedagogical strategy for a given instructional event; finally, thanks to the engine, the system puts together information retrieved from the previous models and delivers the proper content to its final user. In the overall design we should not forget to manage also the interaction between the system and the student. An experienced human teacher knows in fact very well how to communicate his feedback through different ways (orally, in written form, using nonverbal communication). She also knows how to pick the most efficient medium for knowledge flow under the given circumstances (e.g. oral exposition, written instructions, visual aids, etc.). In an adaptive system, this interaction occurs through a final component, known as the communication or interface component [163]. Similarly to how it happens in a real class, communication can take place through different channels (text, dialogue, multimedia, etc.) and the interface allows the user to access appropriate learning materials in the most suitable form and receive feedbacks from the system regarding his or her performance.

We may find ourselves wondering if all this architecture, even though sophisticated, can be enough to allow an artificial agent to mimic, partially simulate or even perfectly replicate the behaviour of a real human teacher. At the beginning of their journey, ITS were envisioned as super-intelligent and all-embracing systems capable of delivering

powerful instruction and managing complex learning interactions [22]. They were, in other words, thought to become not only useful tools for supporting learners and helping teachers, but also able to behave as if they were really human beings. For those concerned with the complexity of the teaching/learning process, this idea is overtly in contrast with the observation that the teaching profession normally requires soft skills that we cannot so easily engineer in a machine. Many people, for instance, would not believe that a cold machine would ever be able to genuinely replicate the empathy of a human teacher who really cares about the student's successes or failures. Nevertheless, in the past, similar visions of an ITS as a caring technology have also been fostered [197]. Scepticism is sometimes exacerbated by results. It has been noted that most ITS used at scale today are much simple if compared to the original and strong vision of a system capable of replacing a human teacher [22]. For example, for evaluating a student's mastery of a certain topic, many systems rely on heuristics that may legitimately seem simplistic to an experienced teacher, such as whether the student is able to achieve three successful attempts [22, 110]. More than just simple, such systems, as Baker says, might even be called Stupid Tutoring Systems [22]. However, one fact remains and instills hope in our research: although ITS do not behave intelligently, they can be designed intelligently (i.e. leveraging human intelligence), so that they can become complementary tools in the hands of intelligent teachers who want to help students to become successful [22]. Borrowing a metaphor from [202], the field of adaptive learning technologies is like a rose garden: its promising rosebud of unopened flowers are perfectly capable of foreshadowing beautiful blossoms, but the blossoming process may need a very long time. This imagery could be applied to the broader field of AI as well, whose history is well known for its oscillation between an initial state of enthusiasm and a subsequent disillusion, which only in recent years researchers have been able to dispel.
The term **Intelligent Learning Systems** (ILS) can be used to refer to those systems that adopt AI methods for knowledge representation and for learning support. A good share of ILSs are both adaptive and intelligent.

The focus of the thesis is the analysis and identification of prerequisite relations in textbooks and, using the same rosebud metaphor, this research issue sometimes is rather like a rose foreshadowing more thorns than blossoms. The fruits of this topic of research are not, we are afraid, immediately appreciable by final users (e.g. students and teachers) of an ILS. But nevertheless they are fundamental for such systems. **Prerequisite relations regard first of all the design and implementation of the**

knowledge model. This model will be therefore discussed in the next section, while **the other components listed and shortly described above will be left aside since they are out of the scope**. Given their unceasing developments, in fact, the other components would probably deserve an entire and different thesis.

1.2 Knowledge Model

As mentioned above, one of the first issues we need to face when we want to create an ILS is making the system able to know a given domain of interest. In other words, the system should "know" which are the knowledge components (e.g. concepts, rules, skills) belonging to the domain and reflected in the learning materials (as well as the pedagogical dependencies between such components). In order to make this knowledge understandable and processable by the system, knowledge components must be organised in a formal structure which incorporates such components and explicitly describes relations between them using a proper language. Knowledge Representation (KR) comes to our aid since this the field of Artificial Intelligence concerned with the use of formal symbols to represent a collection of propositions believed by some agent [132] (in our case the agent would be the intelligent learning system). More specifically, we need to define: a) a formal structure to use; b) the minimum unit of this structure; c) the types of connections between such elements; d) the formal language. A knowledge model can be based on different ideas of what a minimum knowledge component is. A knowledge unit can be more or less detailed (e.g. concept vs topic) and the curriculum can be practically organised according to various formal structures such as hierarchies, semantic networks, frames, ontology and production rules [163]. In the present research, the knowledge model is seen as a set of concepts organised in a network structure, which will be referred to alternatively as concept map or concept graph. Knowledge elements are considered instead to be linked together according to pedagogical sequences expressed by the so-called "prerequisite relation", which states what the learner should know/learn first before approaching a new concept.
Domain or content modelling usually requires the manual or automatic identification of knowledge components and relations. With a manual approach, these tasks are commonly performed by domain experts (e.g. teachers or researchers), because they have enough experience with the domain to know how to build a model from it and they understand the prerequisites dependencies between concepts. These people might be, for instance, teaching that subject for many years.

The knowledge model also serves as a basis for other models, therefore its design and implementation affect the performance of the entire system. In particular, the learner model can take advantage of the information gathered in the knowledge model: it can be, in other words, "aware" of which knowledge elements are to be traced in learner's competence along a certain pedagogical sequence. Knowledge tracing is the process through which a tutor maintains an estimate of the probability that the student has learned each of the elements in the ideal model. Based on this probability estimation, the tutor presents an individualized sequence of materials or exercises to the student, until she has mastered every element [64]. Bayesian Knowledge Tracing is the most popular approach [64, 66, 243], but valid alternatives have been proposed, such as approaches based on Performance Factors Analysis [175] and Recurrent Neural Networks [178]. A knowledge model hence provides a basis for assessment, diagnosis, instruction, and remediation [201]. Changes in a student's knowledge (e.g. learning and acquisition of knowledge components) are essentially changes in her cognitive and brain states: for this reason, they cannot be directly observed or directly controlled, but they can still be inferred from data, as the student performs during assessment events [121]. This is possible since every problem is associated with a set of domain knowledge components. Also learner models can be represented as a graph (as a matter of fact, an imitation of the corresponding knowledge model graph), where each node (corresponding to an element in the knowledge model) has a value that measures the learner's current mastering of the underlying element (see [161, 162] for this design, called overlay model). Therefore, pedagogical dependencies expressed in the knowledge model serve as a guide for administering appropriate educational content for a student given her prior knowledge and current state of knowledge. Assessment through ILS usually consists of a pool of items related to concepts defined in the knowledge model. A common technique for determining a student's knowledge state given her answers to the test items involves the creation of a Q-matrix, i.e. a binary matrix mapping each test item to a set of underlying concepts in the knowledge model [23]. By representing, for instance, in the matrix Q every concept with a row and every item with a column, $Q_{c,i}$ tells us whether concept c is associated with item i.

1.2.1 Concepts

Concept-based learning systems encode components in their knowledge model using concept structures [14]. This approach implies that each knowledge unit refers to a concept, and a specific set of concepts is taken as a basis for modelling and structuring

domain and content knowledge. Depending on designers and users' needs for granularity, concepts can be defined as very general (e.g. algebra, geometry, etc.) or very specific (e.g. integer multiplication, fraction denominator, or even more specific). Moreover, the set of concepts can be more or less populated, depending on how much we want this set to be representative of the full domain of the subject. Granularity and completeness show us very well how the construction of an intelligent learning system is not a trivial issue, and challenges arise even when we must make the first decisions. Not by chance, when we talk about *concepts*, a sophisticated field such as Philosophy easily comes to mind, since it has paid great attention to the topic since ancient times. Abstracting from the objects of the real world, Philosophy has generally described concepts as mental representations denoting an abstract class of things. According to this definition, concepts constitute the building blocks of thoughts that are fundamental in all cognitive processes [46], among which learning and teaching can be rightfully included. Concept learning and concept teaching are crucial processes in education and knowledge acquisition; the term "concept" is often adopted in Pedagogy and related fields with an interpretation that is very much similar to the definition given by philosophers [153, 183]. Despite the effort of leveraging the precious heritage of philosophical and pedagogical thought, studies in Knowledge Representation frequently opt for a more practical interpretation of the term "concept". A concept can be indeed regarded in knowledge modelling as an atomic and discrete component in a knowledge structure (e.g. a concept graph or an ontology), that in turn represents a subset of a domain [100, 164]. An intuitive approach is considering a concept as represented by a linguistic (more precisely, a lexical) entity constituted by a term. According to this perspective, concepts are expressed in texts as keyphrases (both single word or multiword terms), and each term depicts a fragment in a domain knowledge. Thus, the set of concepts in a given domain tends to be equal to the set of terms, i.e. the terminology, of the domain. Terminology, as a discipline, is a branch of Linguistics that investigates the set of specialized words (as well as their associated meanings and inter-relations) related to a specific domain [50, 190]. Rather than being concerned with philosophical or psychological issues involved in the notion of "concept", studies in terminology consider concepts as discrete units in a knowledge structure that represents a specific domain and reflects the current state of knowledge owned by an expert or by a group of specialists [190]. Concepts are therefore represented in the lexicon of a language. In particular, special domain terminologies tend to reduce semantic ambiguity and synonymy for the sake of clarity: this results in the frequently observed phenomenon of concept-term univocity, which implies monosemy of domain-specific terms

(i.e. each concept corresponds univocally only to a denoted concept [96]). Therefore, the lexicon of a subject language reflects the conceptual organisation of the discipline and tends to provide as many lexical units as there are concepts in its subspace [190]. In such a way, a knowledge structure consists of variously interlinked concepts, and the link between concepts and terms is traditionally established by the definition (i.e. a linguistic description of a concept's attributes, which conveys its meaning and often make use of other related concepts) [190].

In some practical applications, a term-based representation of the learning content might not be enough. For instance, in the task of educational resources linking (i.e. automatically find educational documents that deal with the same topic), term-level representations may suffer from the term-mismatch problem, i.e. different keywords could be used to express the same concepts in different resources. In fact, even if specialised languages tend to reduce polysemy and synonymy for the sake of being unambiguous, a mismatching problem can still surface. This happens for example when a generic term occurs as a modifier in a multiword term and thus can be replaced by a synonym (e.g. "document classification" vs "text classification") or when an acronym appears in its expanded version in one text but not in another (e.g. "machine learning" vs "ML"). Another issue associated with term-level representations is constituted by absent key phrases, i.e. terms that are not openly mentioned in the text, but nonetheless are among the topics that contribute to the deep semantic meaning of the text [152]. As opposed to term-based level representations, a document can also be represented at topic-level. In this case, the document is reduced to a probability distribution on a fixed set of topics using algorithms such as latent Dirichlet allocation (LDA) [28]. Topic modelling may be capable of solving cases of mismatching key phrases by extracting the same topic for two or more of such terms. On the other hand, knowledge units provided by a topic-level representation are often too broad to identify reliable and relevant online materials and meet a particular learning need [218]. The choice of representing knowledge components at topic-level or term-level affects both manual annotation and automatic extraction. In the former case, as we said, concepts extraction can be addressed as a task of topic modelling (as in [229], who used Latent Semantic Analysis to extract concepts), while in the latter case it can be addressed as a task of terminology extraction. In the latter case, concepts are intended as the most relevant domain-specific terms that occur in a document, and the relevance is commonly measured with some standard Information Retrieval metric such as TF*IDF (i.e. Term Frequency - Inverse Document Frequency). This statistic gives us an idea of the importance of a term (either a single word or a

complex nominal structure) in a specific document by computing how frequently the term occurs in the document and penalising it if it also frequently appear in a larger collection of documents: the logic behind it is that a domain-specific term should be very frequent in a domain-specific document but not in general-domain texts. In some application it may be worth trying to overcome the limitations of both approaches. [188] presented a knowledge model organized into a three-layered hierarchical structure (term layer, concept ontology and topic taxonomy). In case of document linking, [218] proposed to link educational resources using concept embeddings [111, 206], that were generated by utilising domain specific educational content and external knowledge graph resources.

Speaking of knowledge graph resources, an active area of research deals with the semi-automatic or automatic building of ontologies by extracting concepts from different sources [17, 44, 107, 199, 246]. Concepts constitute a significant part of the knowledge of the world owned by a human being and about which human beings communicate through words [157]. Therefore, ontology learning often adopts Computational Linguistics and Natural Language Processing (NLP) methods to find candidate terms denoting concepts in a textual resource. Pattern-based linguistic approaches employ syntactic parsing to identify domain terms among short noun phrases in the text [81, 97, 112], while statistical approaches discover the degree of *termhood* of words in texts relying on distributional properties [87, 155, 182, 213, 245], or on sentence-level contextual information [58, 59, 69, 227]. The above methods fit quite well also with the problem of automatically discovering educational concepts for building concept maps. In our case (see 5.1.1) we relied on a statistical approach based on linguistic analysis and machine learning to extract a set of candidate terms which we manually revised in order to obtain concepts.

1.2.2 Relations

As happens in relationships between objects in the real world, concept structures in knowledge models can encompass a wide range of links, each one representing a different type of relation between knowledge components. Broadly speaking, we can classify the most common relations in three groups: traditional semantic relations, complex relations, and strictly pedagogical relations. Traditional semantic relations are well studied in an overlapping area of fields ranging from Linguistics (in particular semantics), Knowledge Representation, Object-Oriented Programming (OOP) and Terminology among others. They notably include the generic relationship and the partitive relation. The former establishes a taxonomic hierarchy between two concepts belonging to the same category,

where one is a broader or generic concept (and is called superordinate, superclass or supertype), while the other is a narrower or specific concept (called subordinated, subclass or subtype) [190]. In studies of semantics, this relation is known as hypernym-hyponym relation, while in KR and Computer Science is generally simply called *is–a* relation. On the other hand, partitive relationship expresses a connection between two concepts, one denoting a whole and the other a part of this whole. In semantics this "whole-part" relation is known as holonym–meronym relation, while in fields such as KR and OOP a further subdivision is often made (*has–a* relation when there is an aggregation, *part–of* in case of composition, and *member–of* for containment). Compared to generic and partitive relations, complex relations embrace a wider range of possibilities, therefore they more accurately capture the semantic of a inter-relation between two concepts [190]. Some example can be, just to name a few, "is caused by", "is based on", "is a property of", "is an instrument for", "is a material for". As we can imagine, many of these relations can only exist between certain concepts, thus complex relations are often subject to restrictions depending on their semantics. Lastly, pedagogical relations are traditionally not covered by linguistics or KR, but in intelligent learning systems they are inevitably considered, annotated and possibly extracted, because of their importance for educational purposes. This group includes at least the following relations: prerequisite, co-requisite, is related, is suggested and remedial relationships. Among pedagogical relationships (and arguably among any type of relationships), the most commonly found in intelligent learning systems is the prerequisite relation, representing the fact that one concept must be learned before another. In a concept pair linked by such a relation, the concept that has the precedence is unanimously called prerequisite, while the subsequent can be referred to in many ways ("outcome", "advanced", "subsidiary", even "post-requisite" [1, 239]). Co-requisite relation indicates that two concepts are in a mutual relation, so one is not a prerequisite of the other nor viceversa (they can however be both prerequisites or subsidiaries of a third concept). "is related" is arguably the most general and vague association, subsuming also other types; "is-suggested" can show a link to concepts or resource that are useful for a in-depth reading; "is remedial" brings up special concepts/topics/resources that are accessible by students who need to revise a topic (for example for a makeup test) [209]. Tutoring systems commonly incorporate hierarchical or network-like structures with traditional semantic links *is-a* and *part-of*, since these relations are useful to categorise topics into classes or subclasses, and concepts into more generic or more specific [39, 40]. Complex relations play a significant role in concept mapping activities (see 2.2.1), because they allow students to create a detailed representation

of the conceptual structure conveyed by a text about some topic. Nevertheless, manual annotation of a full concept map including complex relations is very expensive. Concept maps, with both concepts and relations, can be also automatically extracted, and Concept Map Mining (CMM) has been applied to various documents such as academic papers and scientific essays [55, 230]. Given the particular importance of causal relations in many subjects (e.g. History, Medicine, Physics), it may be worth enriching learning materials and knowledge models with an explicit representation of this type of relation. ILS which utilize hierarchical causal concept maps have been for instance proposed for helping anatomy students to handle the complexity of the subject [120].

Above all relations though, the prerequisite relation (henceforth also indicated as PR) is the most commonly included in intelligent systems, since it is the most important or at least the most essential from a pedagogical point of view. This is true to the point that in many ILS it is the only represented relation [40]. Semantically, the PR covers a fuzzy area, partially overlapping with other kinds of relations. We can in fact notice that a prerequisite is frequently a hypernym: if $A \prec B$ (read: A is prerequisite of B), there is indeed some probability that A is also a hypernym of B, because at least in a typical top-down explanation prerequisites tend to be more basic or generic (representing a broader class), while their subsidiaries tend to be more advanced or specific (representing a narrower class). A similar tendency applies to holonym-meronym relation (in a book a new topic can be presented as a whole in the first place, then each of its components can be described) and causal relation (e.g. the explanation of a phenomenon or a historical event is followed by its effects). Moreover, PR shares some space also with temporal relations, and specifically with the precedence (or *before*) relation. This temporal nature of $A \prec B$ can be noticed when between A and B there is an order of precedence, either because they point at different parts in a content sequence, or because they are concepts denoting temporal events. In the former case, $A \prec B$ can be equal to $A\ before\ B$ because authors tend to organise learning content in educational resources so that topics and concepts that are explained before (e.g. in a previous section of a book) are prerequisites of what is explained after. The latter case is particularly evident in a discipline such as History, where what happened before is commonly a prerequisite of what happened later (e.g., Robespierre's Reign of Terror is a prerequisite of Napoleon's assumption of power).

1.3 Why prerequisite relation?

Because of its importance, the automatic identification of prerequisite relationships between concepts has been identified as one of the key requirements for modern, large-scale online education [94, 145, 169, 214]. Pedagogical relations are in fact of great interest in the Artificial Intelligence in Education (AIED) community for automatic construction of domain ontologies and concept maps (CMs) [70]. Prerequisite links can indeed support several adaptation and user modeling techniques [40]. By manually providing or automatically inferring an explicit representation of prerequisites, we can for instance generate a reading list [78, 99], build personalised learning paths [6] or develop recommender systems of educational resources that suggest adequate materials (i.e. with an adequate prerequisite structure with respect of a student's state of knowledge). A manual approach for enriching educational resources with PR can be literally unfeasible, since the manual annotation of PR in texts is time-consuming, and in some cases experts can be hard to recruit. As a result, automatic prerequisite identification is a task that gained growing interest in recent years, especially among scholars interested in automatic synthesis of study plans [6, 11, 95, 240]. While several methods exist (e.g. [59, 229]) to face the issue of automatic concept extraction, the automatic identification of prerequisite relations among concepts is still an open research problem. Interestingly, this contrasts with the fact that there is a long-standing interest on learning sequences and dependencies in pedagogical and instructional theories (see 2.1).

The more general issue of relationship extraction is a well-known task of Natural Language Processing and Information Extraction. The main goal of this task is to identify relations between entities in a document (see [193] and [67] for comprehensive surveys) in order to give a structured representation of the information conveyed by the text. Many types of relationship can be identified, such as temporal [140, 228] and lexical-semantic relations [45], to name only a few. In this line of research [249] and, more recently, [219] retrieves relations exploiting syntactic analysis of sentences in a text and use them to automatically build concept maps. More similarly to our approach described in section 5.4, [130, 242] use burst analysis to recognize relationships between concepts and draw them as links in a concept map. Extracting PR from educational materials is a relatively new field of research. We notice in particular a growing interest towards the analysis of scientific and educational texts [18, 89, 94], especially online digital resources, learning objects and MOOCs (Massive Open Online Courses) [149]. The problem of identifying learning prerequisites between units or concepts across a

curriculum received particular attention in Educational Data Mining literature since the proliferation of digital educational resources, the spread of Learning Objects and MOOCs, as well as the re-affirmation of the textbook as a medium of learning (see 1.4). These scenarios offer both opportunities and challenges to be addressed. Nowadays, learning is not limited to follow a course, but rather takes also the form of finding information independently and in a disorganized way, which may leads to lack of orientation during the learning process. While in the class a student can ask her teacher and hopefully receive a personalised recommendation, in online autonomous learning the problem is greater and the learner might not be feeling satisfied by just asking questions in a discussion forum. MOOCs, on the other hand, pose the problem of tailoring the course according to each students' peculiar backgrounds and needs, avoiding the "one size fits all" philosophy. Curriculum automatic planning or sequencing is the task of finding the next best element (concept, topic, problem, etc.) for students to learn [209]. In this sense, finding a possible ordering between learning resources in a MOOC by means of automatic identification of prerequisites can represent an important step towards the development of learner-centered courses [146]. MOOCs could be enhanced with a concept map and a sequencing algorithm that acts as an academic advisor: this would also avoid course overlapping, allowing students to skip repetitions and guiding them across MOOCs provided by different institutions or hosted in different platforms. On the other hand, the limits of MOOCs in their collaborative functionality, despite their initial promise of gathering learners from all parts of the world, have been noticed in [104], who presented a graph-based infrastructure design which enables instructors to run social activities leveraging orchestration graphs [71] (a framework for modeling pedagogical scenarios).

1.4 Example of application of PR in textbooks

The research presented in this thesis aims at giving contributions to the study of prerequisite relation in fields related to Artificial Intelligence in education. This topic of research, as we explained in the previous section, is potentially beneficial in different application contexts. As will be clearer in the next chapters, though, we show a particular interest towards textbooks. Discussing the use of textbooks in ILS in 2020[2] may appear

[2]As he was writing these lines (January 2020), the author could not be aware of the imminent COVID-19 outbreak, which even at the present moment (June 2020) is giving a great impulse to e-learning environments and is pushing a multitude of teachers and students towards the use of digital learning resources, among which digital textbooks. This fact must be probably taken into account in addition to the

outdated, especially if we consider that interactive textbooks are among the oldest forms of technologies in adaptive and personalised web-based learning [195, 237]. Adaptive textbooks were indeed a popular topic of research during the 1990s. The spread of World Wide Web, with its hypertextual nature, suggested in fact the use of textbooks as the most natural choice for conveying online learning materials. After all, as philosophers such as Barthes and Foucault noticed, traditional paper books had already an implicit hypertextual structure, conceivable in terms of networks and links to other parts of the same book or to other books [127]. Since then, many technological advances have been certainly achieved in the last decades within the field of educational technology, and other formats or learning environments appeared (e.g. Learning Objects, MOOCs, ITS embedded in MOOCs [7], etc.). Nevertheless, textbooks (in their traditional or digital form) still represent a major source of knowledge for students in online and blended learning. They are still, for instance, largely sold in real and digital stores, and they are used at different level of education all over the world. The use of textbooks is also recommended in learning environments where they officially do not play a central role. A common MOOC, for instance, typically provides to its users a learning experience mainly focused on didactic video-lectures, with some peer-reviewed activities or discussions in forums, and very few reading materials (e.g. not textbooks). The way of perceiving MOOCs, i.e. getting rid of textbooks and focusing nearly exclusively on watching videos, may not necessarily be a fruitful learning strategy. Additional textbook reading in online courses could for instance help students to learn more [49].

The importance of textbooks is intensified by the process of digitization, which has made accessible a large amount of digital resources to be used as a form of learning material. This fact suggests a usage of web resources according to a "web as a textbook" paradigm, i.e. it transforms the web of rich but chaotic educational materials into an adaptive, web-scale textbook, where users-learners can be guided by some intelligent agent into the most relevant pages according to their knowledge and needs [125]. This vertiginous scenario even opens the door to suggestions coming from some of the most fascinating literary visions, such as the hypernovel *Rayuela* [65], where a disoriented reader is guided through a complex textual content thanks to a set of instructions provided in the form of sequences (that do not preclude a free exploration of the content), or Borges' *Library of Babel* [33], where an expanding universe of bookshelves offers to its visitors all the possible permutations and people look for an elusive super intelligent man who possesses all the knowledge (and thus would be capable of providing a perfect

motivations discussed in the body of the text.

path through the library's content). As a result, intelligent textbooks nowadays represent a long-standing and yet reappearing topic of research, that poses new challenges, both of theoretical and practical nature. Undoubtedly, the most elementary approach to e-book is just a digitalised copy of a paper book. However, authoring systems have been presented to allow instructors to enhance static electronic e-books by incorporating multimedia and interactive components (e.g. multiple-choice questions) or augmenting them with external links to multimedial resources available online [156]. Authoring tools have also been proposed in conjunction with text-mining methods for automatic skill model discovery and annotation of textbooks [147]. Reading analytics can trace the reader behaviour: learners behavioral patterns can be for instance constructed by analysing their reading logs [241], while their interaction with the textbook can be used to infer the current state of student knowledge [217]. Intelligent textbooks can also collect and fuse together data taken from different sources and of different nature: time series (e.g. time spent on a page), reading speed, reading sequences of reading (pages or sections), learner's notes, data taken with sensors (camera, eye trackers, microphones), the reading strategy of the learner (intensive reading, scanning, skimming) [34]. Others studies proposed multiple textbook integration for increasing the coverage of a domain [9] or textbook linking across different languages for helping university students to read textbooks in a foreign language supported by on-demand access to relevant reading material in their mother tongue [10].

We believe that PR identification is closely related to at least two topics of research among those associated with the re-emerging field of Intelligent Textbooks (see [207] for a more detailed list, from which we took the following excerpts). These two topics are:

- Textbook modelling. In particular, we refer to how we can study, analyse and automatically identify prerequisite relations, besides the general semantic structure embodied in a textbook, with the aim of enhancing its readability and providing intelligent functionalities to its users.

- Textbook augmentation and knowledge visualisation. We refer to how we can enrich textbooks with explicit representations about its conceptual dependencies, as well as how we can communicate such knowledge by means of visualisation techniques, concept maps or other forms of knowledge-rich representation.

The first issue represent the central research topic of the thesis, discussed even starting with this chapter. The second issue constitutes a further track within studies on intelligent textbooks that can be associated with prerequisite relations, since concept

maps are a natural way to visualise such relations. Besides, more in general, various forms of knowledge visualisation may take advantage from a deeper concept dependency structure to provide rich content to users.

1.5 Guide to the thesis

The rest of the thesis is organised as follows:

- Chapter 2 will discuss the prerequisite relation from multiple points of view. Given its multidisciplinary nature, the chapter will give an account of different perspectives both in pedagogical sciences and in formal Knowledge Representation. Moreover, the chapter will present basic approaches that can be employed to automatically identify this kind of relation, the literature concerning the annotated resources with prerequisite relations, and the related prerequisite-enriched datasets.

- Chapter 3 will discuss questions that have arisen during the investigation of this topic of research, and introduces the framework that we conceived to address such questions.

- Chapter 4 will describe the architecture of the framework that we propose as a support for the community of researchers working on the identification of prerequisite relations in textbooks. We conceived this framework as a comprehensive and coherent multi-modular environment that allows its users to perform all the major tasks involved in PR research issue, therefore each of its core modules is described. Among them in particular:

  - modules to support the **manual annotation of datasets** with prerequisite relations,
  - modules for the **automatic extraction** of prerequisite relations from textbooks.

- Chapter 5 will deal in detail with the development of the framework and includes in addition:

  - the definition of a method and guidelines for the manual annotation of prerequisite relations in textbooks,

  - the experimental evaluation of the methods for the extraction of prerequisite relation. In particular, it will focus on the use of **manual annotated dataset for evaluation purpose**, i.e., used as gold standard.

- Chapter 6 will finally draw conclusions, discuss limitations and future possible developments.

CHAPTER 2

THE PREREQUISITE RELATION

As noted by [113], in the design of ILS (see chapter 1) the term "prerequisite" seems to bear at least two meanings: on the one hand it may express a pedagogical relationship between two elements that the student should learn, on the other hand it may indicate a formal mechanism that can be used to partially order two units of instruction (concepts, pages, exercises or similar) inside a sequence of learning materials. In the former case, the use of the term "prerequisite" is justified by some educational theory or some argumentation rooted in the field of cognitive psychology, while in the latter case the use of "prerequisite" as an ordering principle is generally motivated by instructional and sequencing purposes. Pedagogists and Instructional Design theorists have indeed struggled to provide convincing frameworks that include a description of what a prerequisite is and how we must deal with that when we need to build learning paths. On the other hand, the engineering of prerequisite relations for building intelligent learning systems is largely a matter of Knowledge Representation, i.e. a field of Artificial Intelligence which may demand simplification or assumptions in order to formalise this relation into an unambiguous (e.g. mathematical) model (such as a graph or an ontology). As already stated in chapter 1, as the amount of open digital resources is massively growing, the human annotation of every single text has become plainly unfeasible. As a result, the current situation arouses the interest of researchers towards the challenging task of automatic identification of prerequisite relations. In this lines of research, the development of efficient methods of automatic extraction is arguably the ultimate goal, while the construction of annotated datasets becomes a precious undertaking for training

and evaluating machine learning algorithms.
The present chapter addresses all these issues and is organized as follows: section 2.1 will provide an overview of some of the most significant theoretical discussions emerged in fields of educational psychology and instructional design about prerequisite relations and sequencing principles; section 2.2 discusses how the prerequisite relation can be found engineered in knowledge representation for courseware purposes (e.g. domain/textbook modelling in ILS, automatic testing and lesson planning, etc.); section 2.3 recollects strategies commonly employed to automatically extract prerequisite relations; lastly, section 2.4 provides a brief survey of the datasets that include a human annotation of prerequisite relations.

2.1 The prerequisite relation in learning and instructional theories

This sections concerns pedagogical theories involving prerequisite relations. We will not give however a comprehensive and complete listing of all theories, since this would be out of the scopus in the present work. Rather, we will give a reasoned overview of some of the most significant cases with respect to the topic of the thesis. The aim of the section is to help the reader to better understand how the issues of prerequisite, prior knowledge and curricula sequencing are dealt in this field.

2.1.1 Prerequisites in Behaviorism

An early statement on the beneficial conditions needed by a student to assimilate new knowledge was expressed in the first half of the twentieth century by the behaviorist psychologist Edward L. Thorndike. According to his "Law of readiness", formulated in 1913 [221], a satisfying state of affairs can indeed arise when a person is ready to learn and he is allowed to do so. Although the preparatory adjustment required to create such a positive condition embraces a wide range of physical, mental and emotional factors, Thorndike's law has also been interpreted in terms of prerequisites, suggesting that if a student does not possess prerequisite knowledge or skills, then most of his attempts to learn are not rewarding and less effective [194]. Addressing the issue of curricular sequencing, Thorndike asserted that a skill should be introduced when it is most facilitated by the immediately preceding learnings and when it will most fully facilitate the immediately following learnings [194, 222]. Consistent with

his law, Thorndike predicted in 1912 an instructional application represented by a smart and interactive textbook capable of automatically showing material in a logical order, so that its pages would become visible only to students who had done enough to understand them [220]. Thorndike's intuition became a reality with the construction of Pressey's and Skinner's teaching machines [180, 204], i.e. mechanical devices that administer learning materials and multiple-choice questions. Besides the well-known theory of operant conditioning, B.F. Skinner is therefore generally credited also with the development of Programmed Instruction (PI), an instructional approach based on the idea to organize learning activities as a progression ("chaining") of small and discrete steps according to a specific behavioral objective and taking into account the learner's entry skills. Learning sequences can be then structured as linear programs (i.e. all students proceed through them following the same order, but not necessarily at the same rate) or as branching programs (i.e. students' transitions depend on how they perform during the tests). A certain degree of personalisation can be achieved allowing students who learn quickly to skip some steps and bypass much of the repetitive items contained in linear programs, whereas slower students will undergo an appropriate schedule of additional reinforcement, in a typical behaviourist manner, until they can reach the learning target state thanks to a gradual approximation ("shaping") [73]. A much clearer definition of what the student must learn and know become possible with the introduction of Bloom's Taxonomy [31], a hierarchical model that can be used as a rubric for systematically classifying learning objectives into a stack of increasing levels of complexity. In the same years John B. Carroll [48] formulated a quasi-mathematical model of school learning where the time spent by the student represents a significant variable that underlies differences in learning achievement. A lack of good quality in instructional design, e.g. lacking a sequencing principle , increases the time needed for learning. An early application of many of these ideas is found in Fred Keller's instructional plan known as Personalized System of Instruction [116], where learning materials are broken down into distinct units and several kinds of relationships are traced between such units. As an essential relation, one unit generally provides prerequisites for understanding the next unit, which in turn provides a deeper elaboration of the contents of the proceeding unit (in such a way that, intuitively, as a criterion of advancement, students must satisfy a "unit mastery" condition in one unit before proceeding to the next). This principle also constitutes a major element of Mastery Learning, formalized by Benjamin Bloom in 1968 [29, 30], where students have to master a minimum percentage of prerequisite knowledge (e.g., 90%, as resulting from an assessment test) before moving forward to

learn subsequent materials.

2.1.2 Prerequisite relation in Information Processing Theory

In contrast with the externalist perspective of Behaviorism, several ideas that can be ascribed to the Information Processing Theory (from 50s to 70s) emphasise the role of the human mind as an information-processing system. From this point of view, human cognition is thus seen as a series of mental processes, and learning as an acquisition of mental representations [148]. In these mental processes a major role is played by what the student already knows. His or her prior knowledge is indeed, in David Ausubel's opinion, even "the most important single factor influencing learning" [21]. Ausubel proposed a theory in which "meaningful learning" is seen in opposition with rote memorization [19]: in order to achieve the former, students must link new pieces of knowledge (concepts and propositions) to their prior knowledge by means of what Ausubel called "advance organizers" [20], i.e. a set of cognitive instructional strategies (e.g. maps, hierarchical representations, recall of previously explained concepts, etc.) that are used to facilitate learning and retention of new information in long term memory and to encourage students to find connections between new and previous materials. A major contribution to the study of prerequisite relation as a criterion for curricula sequencing can be found in Robert Gagné's pioneering work on learning hierarchies. In his theoretical framework Gagné proposed the notion of "conditions of learning" [91], i.e. the circumstances that must be satisfied to allow a person to learn new skills or knowledge. Instructional designers must clearly specify every internal or external condition, among which prerequisite concepts and skills constitute significant internal conditions that are required to cognitively process new knowledge and must hence be activated from the student's long term memory by means of instructional strategies included in learning materials and teachers' explanations [93]. Prerequisite concepts and skills can form a full set of capabilities having an ordered relationship between each of its components. This interrelated structure results in what Gagné called a "learning hierarchy" [90, 91], that can provide a valuable guide for building study plans with an effective disposal of learning components. An ordered relation between two capabilities implies that there is a subordinate learning task that can generate a substantial amount of positive transfer for learning a new knowledge, skill or task [92]. Interestingly, in Gagné's original formulation a learning hierarchy should not necessarily be interpreted as the only possible learning path leading to the final outcome, nor as the most efficient path for every student, but rather as the most probable expectation of the greatest positive transfer for a given student, who may

nevertheless be able to skip some parts of an adaptive program of instruction, if he or she already possesses certain skills [92]. As claimed also by [176], an optimal sequence of content presentation, *per se*, is not necessarily correlated to the learning result. As a matter of fact, even a highly disorganized sequence (i.e. lacking any ordering) may not entirely prevent a learner from reconstructing a “coherent and meaningful internal arrangement” [92], even though this procedure could probably take extra time and efforts, and many learners may not reach an optimal learning or might drop the course because of frustration. The advantage of the learning hierarchy is hence to identify prerequisites that should be achieved in order to best facilitate the learner at every level. The strictly hierarchical nature of Gagné’s model has on the other hand a limit, since it is hardly suitable for representing learning prerequisites in areas where a rigid hierarchy is not the case [183]. This drawback encouraged many authors to shift from a tree-like or hierarchical representation to a network or graph formalism [173] . While learning hierarchies assume that only a relation (i.e. the learning prerequisite) is sufficient to describe the entire leaning content, network-based approaches might bring to represent a large number of detailed relations, most of which are of little value for instructional design purposes [183]. For this reason, [183] specified a small set of relations that are of great utility for instructional designers that aims to select, sequence and synthesize the subject matter components (single concepts, principles or facts). Four fundamental types of content hierarchical structures are discussed (plus one extra relation which is not hierarchical in its nature):

- Learning hierarchy, which can be intended as a synonym for Gagné’s learning hierarchy and it is based on the "learning-prerequisite relation": it describes what must be known (what the learner must be able to do) before something else can be learned.

- Procedural structures, which are based on "procedural-prerequisite relation" or "procedural-decision relations": the former relation is present between the ordered steps of a single procedure (i.e. the performer must do X before approaching Y), while the latter is present in procedures involving alternate decisions, as for instance the different steps of the hypothesis test used in statistics.

- Taxonomic Structures, i.e. concept X is a kind (*is–a*), or is a part (*part–of*) of Y.

- Theoretical Structures, which are basically chains of causal relations among concepts.

- Lists, i.e. concept structures that actually have not a hierarchical nature, but they may still show some kind of ordering among their elements (e.g. rocks listed in order of hardness, historical events in chronological order, countries in order of size). Given a list of elements, there may be many criteria to order its members.

2.1.3 Prerequisite Relation in Constructivism

The main metaphor conveyed by most Information Processing theories (i.e. the learner's mind as a computer including a central processor, sensors, memory registers and so on) can be considered a bit narrow if judged from a constructivist perspective. According to such perspective, learning is not only the result of a transmission and procession of atomic pieces of information from one source (the teacher's mind) to a storage (the student's mind), but it also involves an active effort of knowledge construction, inductive discovery and sense making performed by the learner during an interaction in a social context. Using a metaphor proposed in [128], teacher and learner's minds, rather than two computers, constitute something similar to an electrical transformer, in which knowledge cannot be physically transmitted but only induced from the teacher knowledge to the learner's circuits containing his previous knowledge [223]. Besides being a philosophical and psychological theory of education, constructivism has also influenced teaching practices and instructional designs, although incorporating constructivist principles in real classes is not always an easy task for educators [148]. Constructivist teachers typically favour the idea of a teacher playing the role of facilitator or mentor (rather than instructor), thus they frequently adopt a minimally guided instructional approach, which has been alternatively referred to as problem-based, inquiry, experiential, discovery, or, more simply, constructivist learning [118]. All these methods of teaching involve that the student discovers and constructs knowledge by formulating and testing a hypotheses rather than just passively reading or listening to the teacher's exposition [118]. This act of discovery is basically a form of inductive reasoning, since students start from a specific example or case and then move to formulate general rules, concepts or principles [38]. In such case, therefore, understanding a specific case is often a prerequisite for effectively learning the general concept. Since in similar contexts the learning process is less guided by a direct explanation, teachers and instructional designers must pay great attention to prerequisite knowledge and therefore assure that their students possess a satisfying background preparation (e.g. declarative, procedural, and conditional knowledge) in order to accomplish the activities [194]. A good domain knowledge is generally a prerequisite for problem-solving tasks, inquiries and debates, because under such activities a student

lacking a decent understanding of basic knowledge associated with the problem is not likely to perform well. Teachers can hence provide scaffolding by carefully posing strategic questions and giving suggestions on how to solve the problem. In particular, a preliminary and careful structuring of materials is beneficial when students are not familiar with the discovery procedure or require extensive background knowledge [194]. Discovery can impede learning when students have no prior experience with the material or enough background information [224]; in such cases, alternative strategies, such as using worked-examples (that are more in line with cognitive load theory), have been found more useful than discovery learning [224]. Constructivism also paved the way to an idea of spiral curriculum in which students confront a great and important topic in more units across the curriculum and from multiple perspectives [194], breaking the rigid sequentiality where each concept is a prerequisite of the next concept. Finally, constructivist learning is frequently associated with peer tutoring and cooperative learning, since these social learning practices are consistent with Lev Vygotsky's notion of zone of proximal development (ZPD). This notion describes "the distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem solving under adult guidance or in collaboration with more capable peers" [232]. This perspective differs from Piaget's theory, another notable constructivist theory according to which Knowledge develops through the individual cognitive activity and follows a generally predictable and fixed sequence [177, 194]. Consistently with ZPD formulation, learners with different background skills and knowledge can cooperate, and each student's learning goals should be defined not only considering what she could achieve individually. Scaffolding provided by a peer or a group can help a student to go beyond the limitations of her current level. The tendency of constructivists to enlarge the notion of learners background, that should not be limited to their knowledge but also includes their social and cultural background, is deeply rooted in the research of a colleague of Vygotsky, Aleksandr Luria, conducted on illiterates from rural areas [144].

2.2 The prerequisite relation in knowledge representation

Knowledge Representation (KR) is a field of Artificial Intelligence whose essential goal is the description of a state of the world using some kind of formal language so that a machine or a rational agent can perform reasoning and computation on it [36]. Given a

knowledge component Y, the PR relation represents, from a cognitive and educational perspective, what a learner must know or study (i.e. another knowledge component X) before approaching Y. More formally, the PR relation can be hence defined as a dependency relation expressing precedence between two knowledge components: in mathematical notation, we will write $X \prec Y$. Despite the fact that we may easily agree with this definition, the prerequisite relation can be represented and encoded using different models. We may wonder then what is the right model to use. Echoing the words of the statistician George Box, we want to remind ourselves that all models are wrong, but some are useful [35]. A model is in fact a useful approximation of the world, more or less relying on some simplifying assumption that we need to take in order to make advancements, because as Paul Valéry once said, what is simple is always wrong, but what is not is unusable[1]. In the next section we will therefore describe what we consider as the most usable or useful models for formally representing prerequisite relations. We are well aware that, especially from an educator's point of view, all these models can create the impression that we are oversimplifying the student's cognitive dimension or that we are unable to really capture the full spectrum of what a prerequisite relation is (linguistically, cognitively and pedagogically).

2.2.1 Prerequisite concept maps

Concept maps (CMs) constitute a straightforward approach for structuring the knowledge components contained in an educational resource. Concept mapping technique was developed by Joseph Novak in the late sixties [167], under the influence of Ausubel's theory of advance organizers (see 2.1.2). Novak proposed concept mapping as a powerful technique to actively build knowledge by incorporating new and already acquired concepts in a complex graph structure enforced by explicit relations between its components [166]. In line with a constructivist perspective of the learning process (see 2.1.3), concept mapping procedure supports the learner, by means of a graphical support, during the cognitive process of creating, representing, explicating and sharing new ideas. Besides their potential in supporting the learning process, CMs can also be a valuable tool for automatically generating lesson plans [4, 134] and evaluation tests [248]. From a Knowledge Representation point of view, CMs represent key concepts of the subject matter and organize them in a formal structure (i.e. a graph) by means of semantic relations. This results in hypergraphs with typed n-ary relationships among concepts, including the

[1] Ce qui est simple est toujours faux. Ce qui ne l'est pas est inutilisable.

prerequisite relation. In an ILS prerequisite links can often constitute the only expressed relationship between concepts. Nevertheless, richer forms of knowledge graphs can also be incorporated, if necessary. In similar, more complex, systems, the set of relationships usually covers also traditional semantic and taxonomic relations (i.e. "is–a" and "part–of" [40]), as well as any other kind of complex or pedagogical relation that can be useful (e.g. "is–suggested", "is–based–on", "is–related–to" and many more). Since each node in a CM represents a concept , both manual and automatic construction of a CM demands a clarification about what we mean with the notion of "concept". We provided discussions concerning this issue in 1.2.1, even though the meaning of concepts and their automatic extraction from texts do not represent the main focus of this thesis. Concept Map Mining (CMM) is the task of automatically discovering concepts and relations in an educational text. Its pipeline implies essentially two sub-tasks: Concept Extraction and Relationship Extraction [229]. Regardless the method of concept representation/extraction, prerequisite CMs have the advantage to visually disclose the formal properties of the PR relation. By definition, the main properties of a PR relation are the followings:

1. binary relation: it always involves a pair of concepts;
2. anti-reflexive relation: concept X cannot be a prerequisite of itself;
3. transitive relation: if $X \prec Y$ and $Y \prec Z$, than $X \prec Z$ (see for instance the CM of Figure 2.1: *browser* $\prec$ *HTTP* and *HTTP* $\prec$ *WWW*, hence *browser* $\prec$ *WWW*;
4. anti-symmetric relation: if concept $X \prec Y$, than $Y \prec X$ must not hold (in the map in Figure 2.1, *network* $\prec$ *internet*, so *internet* cannot be prerequisite of *network*).

These conditions also imply that a prerequisite CM is a directed acyclic graph (DAG).

Given the semantic and the properties of its edges, a set of nodes are of particular interest in a prerequisite CM: source nodes and sink nodes. The former are nodes with zero indegree and in this thesis we will refer to them as Primary Notions, since they represent concepts that are the entry points of a learning path and therefore identify one of the absolute prerequisites (i.e. the concepts that a student must know before attending a given unit of learning or reading a given resource). All other nodes represent Secondary Notions, i.e. concepts that will be somehow explained in the learning material; sink nodes, i.e. nodes with zero outdegree, represent a subset of these nodes and express the Outcomes of a material. Obviously, a Primary Notion cannot be also an Outcome or viceversa, i.e.:

$P \cap L = \emptyset$

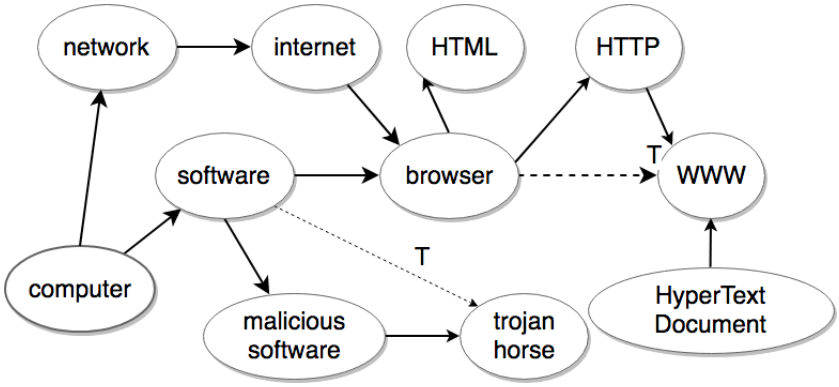

Figure 2.1: Example of a PR concept map (sample from the dataset described in 5.1.1)

2.2.1.1 Knowledge spaces

A prerequisite-enhanced graph-based model can also be found in Knowledge Space Theory (KST), developed by Doignon and Falmagne [72]. This theory provides an elegant mathematical model for representing a learner's competence in a given domain at a given instant of time. KST model is basically a combinatorial network where each node is potentially able to represent a possible subset of concepts in a subject domain. The student's current competence can be then identified with the node in the network that contains the collection of concepts (or skills, facts, etc.) that the student has mastered [79]. The set of nodes reachable from the current knowledge state constitute the outer fringe of the student knowledge. Every node in this fringe is known as Learning State and reveals the concepts (skills, facts, methods) that the student is ready to learn [80]. The learning sequence represents a possible linear ordering of the nodes in the Learning Space for a given student [79]. In KST prerequisite relationships can be formalised by means of the so-called Surmise Relation: two items A and B are in a Surmise Relation if by knowing that a student masters item B, we can surmise that this student also masters item A. From a mathematical point of view, a Surmise Relation is then a partial order on the set of items.

2.2.2 Prerequisites vs outcomes separation

An alternative approach for encoding prerequisite knowledge in educational texts is used, among others, by [42, 53, 125]. This approach aims to distinguish whether a concept can be directly learned from the text (and thus it represents an "outcome") or not (and in this case it is a "prerequisite"). The former term denotes the pedagogical goals of the learning

material [42], but it can also be intended as denoting concepts whose definition can be found in the material [52]. We may note however that defined concepts and pedagogical goals are not necessarily the same thing, because an author can still recall in a given unit of learning the outcome concept of a previous unit by re-providing its definition (e.g. "Remind that, as said in previous section, an array is a data structure that..."). In prerequisite versus outcome separation (P | O henceforth), the problem of manually annotating or automatically identifying prerequisite relations is approached as a binary classification task. Contrary to a network-based approach like CMs, in P | O labels or types are assigned to the concepts and not to the relations/edges. As a result, P | O lacks the rich representation of a full graph that is derivable from a prerequisite CM. This means that, below a textual unit that was annotated with P | O (e.g. a section), we cannot easily understand which concept is strictly necessary to learn a certain outcome. On the other hand, this way of representing prerequisites works well at a higher level of granularity (e.g. section or chapter level) and this can be enough when a student needs to know which concepts she should know before starting to read a text and which concept she will have acquired at the end of her reading. The full conceptual and textual flow that brings the reader from the entry concepts to the final outcomes is probably better reflected in a graph that can be used to represent connections at paragraph or even sentence level. Hybrid forms of representation, i.e. both sentence-level and section or document level, can also be found in literature. [188] used for instance a two-level classification with four types of concepts. At the level of the sentence, a concept can be:

- a defined concept, if it is defined in the sentence;
- a used concept, if it is mentioned in the sentence to explain another concept.

On the other hand, at the level of the document, a concept can be:

- a prerequisite concept, if it is included in the set of used concepts;
- an outcome concept, if it is included in the set of defined concepts.

Note that both in PR CMs and P | O the word "outcome" is often used with a simplifying meaning. From the point of view of teachers and instructional designers, a common good practice is in fact defining outcomes in terms of learning objectives, using rigorous frameworks such as the Revised Bloom's taxonomy [124]. In other words, watching an educational text annotated with just "outcomes" labels, one could ask whether the student, at the end of her reading/studying, will just remember or she will have

understood the outcome concepts. As a matter of fact, we might even wonder if she will be able to apply, analyze, evaluate or create. Objectives in Bloom's taxonomy are indeed organised in six levels and, in order to master skills belonging to a certain level, it can be necessary to master skills at lower levels. This layered hierarchy of learning outcomes is definitely harder to represents than the simpler sink node (as in PR CMs) or binary classification (as in P | O) based models presented so far. [74] proposed a model to represent the content of a curriculum for an ITS by using Bloom's taxonomies and then using several categories of links between learning objectives: compulsory prerequisite, desirable prerequisite, pretext (i.e. weak relation) and constituting. They also decided to simplify these distinctions and only kept a general prerequisite relation, but interestingly they tried to preserve categorical differences derived from Bloom's taxonomies and thus defined semantic restrictions so that, if an objective O_1 is prerequisite of O_2, then in theory O_1 should belong to a lower category than O_2. However, they actually found hard to determine the categories of O_1 and O_2 (for example, it could happen that synthesis exercises are necessary for the comprehension of another element).

2.2.3 Levels of difficulty

A third way to express a prerequisite relation between A and B is by just declaring the difficulty of both A and B: if B is more difficult than A, then $A \prec B$ could be the case, because students are ready to learn the difficult concepts only after they have acquired the simple ones. As in P | O separation, here we do not add classes to relations/edges in a graph but to concepts/nodes. For instance, [56] introduced information on prerequisite relations in a Bayesian Network (BN) based learner model by considering three different levels of difficulty of knowledge items. The resulting configuration is a network where a hypernode L_1 containing concepts $\{C_1^1, C_1^2, ...C_1^n\}$ is a prerequisite of a hypernode L_2 containing concepts $\{C_2^1, C_2^2, ...C_2^m\}$, and $L_1 \cap L_2 = \emptyset$. Depending on our needs, this model can suffer from some limitations:

- Because of the lack of expressed relations between the concepts that are comprised in the set of nodes of each level, we cannot understand which concepts of level L_i is strictly needed for understanding a certain concept in the higher level L_{i+1}.

- Level transitions (i.e. student's promotions) may demand simplifying choices, for example setting a threshold of concepts that the student has to master before going to the next level (e.g., three concepts, as in [56]). While this can works in certain

cases, it can also leave behind concepts that could be later required, pushing us to introduce at least some kind of inter-level relations between specific concepts.

- Sometimes two (or more) concepts in a level can be both (or all) necessary to go to the next level of difficulty, and the lack of intra-level relations do not make visible such independences.

A more complex interpretation of difficulty level is found in [47]. This also extended a BN based learner model by adding a specific layer for representing prerequisites, but in this case the network computes estimations based on the difficulty of the knowledge component (low, medium, high) as defined by the teacher when constructing the knowledge model. This linguistic value is then converted into a probability d that represents the difficulty of the knowledge item in itself, i.e. given that all its prerequisites are known. The meaning of the prerequisite relation $A \prec B$ is therefore intended as a conditional distribution of A and B. This model can be also extended for the case of a set of two or more prerequisites, by using AND/OR gates (depending on whether all the prerequisites are needed for understanding an element or there are two or more alternative ways of getting to know it).
A final question can be posed on how these different ways to encode PR in texts affect the annotation, e.g. whether they are easy to understand for annotators, time-consuming or error-prone. Since in this work we rely on the first form of representation, in chapter 5 we will report the difficulties encountered by humans during the effort of annotating educational text with PR at fine grained level (i.e. paragraph/sentence) and using a PR CM representation (i.e. annotating PR relations between pairs of concepts).

2.3 Approaches for the automatic extraction

As already mentioned in 1.2.2, the automatic identification of prerequisite relations among concepts is still an open research issue, even though there is a long-standing interest on learning sequences and dependencies (at least since Gagné's work on learning hierarchies in the Sixties, see 2.1.2). Several approaches have been proposed to extract prerequisite relations from various educational sources [3, 41, 99, 114, 136, 137, 231, 236, 240]. Broadly speaking, these approaches for PR extraction may rely on some external structured representation of knowledge (such as Wikipedia [214] or other knowledge bases [171]) or may leverage internal linguistic information contained in the document

[99] (often, though not necessarily, enriched with external knowledge as in [136, 236]). In this section we propose a classification based on this criterion.

2.3.1 Approaches based on external structured knowledge

External knowledge bases often provide great sources that can be harvested in order to find candidate prerequisite relations. For practical reasons, we will dedicate an individual section to lexical resources on the one hand and to Wikipedia+Ontologies on the other hand. Finally we will discuss a metric that, even if originally applied using external knowledge bases, it can be generalised to internal text-based approaches as well.

2.3.1.1 Lexical resources

As mentioned in 1.2.2 and 2.1.2, we can expect that prerequisite relations may be overlapping with semantic and taxonomic relations (e.g. hyponymy and meronymy). In educational texts, in fact, when a concept is explained, students often should know the concept denoting the generic class. In some cases, the relationship identification is made even easier by the hierarchical lexical relation between terms representing concepts: *network* and *Local Area Network*, for example, are not only tied by a hypernym-hyponym relation , but they also share the lexical head. This observation suggests that a prerequisite graph can be partially constructed by extracting semantic hierarchical relations. Lexical databases such as Wordnet and BabelNet are commonly used as external resources for extracting candidate hypernyms and meronyms [45]. Semantic relations can be also extracted without external resources (i.e. directly from texts) by means of syntactic patterns (see 2.3.2.1). Alternative methods for automatically discovering hierarchical structures of concepts linked by a hypernym–hyponym relations include word embeddings [88] and distributional semantics [122]. Relying exclusively on one external lexical database can be not always satisfying in terms of precision or recall. In fact, since the external lexical resource was not built from the text under examination, extracted hypernym-hyponym relations can be lexically valid but not really expressed in the text. On the other hand, it can also happen that relations may not be extracted since the resource does not properly cover the domain of the document. Furthermore, the direction of a PR always depends on how this is instantiated in the text, while an external resource is untied from our specific text. Therefore this can bring to the extraction of a candidate relation where both concepts are correct, but the hierarchical top-down direction expressed by the hypernym-hyponym relation is not reflected in the

text, where these two concepts are explained with a bottom-up style (i.e. the specific case first, then the generic concept).
In the same way, non-hierarchical lexical relations cannot be always useful to exclude the existence of a prerequisite pairs. As an example, think about co-requisites, i.e. concepts that have a non-hierarchical nature and are usually presented together for providing complementary knowledge. To clarify, imagine a possible description of the HTTP protocol: the *client*, who sends a request to a *server*, should not be a prerequisite of *server*, nor vice versa. Nevertheless, this way of structuring concepts is consistent with an ontological approach, while in actual texts we might also find authors that explains concepts in such a way that ontologically $A \not\prec B$ and $B \not\prec A$, but textually $A \prec B$ or $B \prec A$.

2.3.1.2 External Knowledge Graphs

Besides lexical resources, other forms of external structured knowledge can be mined to extract PR candidates between concepts in our text. In this sense, ontologies and Wikipedia are excellent sources for deriving structured knowledge. Ontology is a notion that originated in philosophy, but in Semantic Web and Knowledge Representation it usually denotes a formal description of the conceptual structure belonging to a given domain [101]. In particular, an ontology defines properties for its entities as well as semantic relations between them using formal languages such as RDF and OWL. Given the presence of semantic relations, an ontology can thus resemble a taxonomy or a semantic network [101] and also a concept map. Domain ontologies can be used to generate concept maps disclosing the knowledge structure acquired by an individual or a group of learners [129], and in e-learning they can be used for curriculum building, content sequencing [57], Learning Objects (LOs) ordering using Bloom's Taxonomies [200], curriculum data enrichment [102]. Mining content from Wikis has also been frequently proposed as a strategy for enhancing learning, enforcing student's engagement and enriching learning materials [63, 172]. DBpedia is a database that stores structured content from Wikipedia pages and allows the user to make semantic web queries (through query languages such as SPARQL) in order to extract relationships and properties of Wikipedia resources, conveniently expressed in RDF format. [146] presented a machine learning approach for measuring prerequisites between concepts in MOOCs: different binary classifiers have been evaluated based on a set of features extracted from DBpedia. [145] explored the DBpedia Knowledge graph to find candidate concepts, then implemented a pruning method to reduce the set of such candidates and finally employed a supervised learning algorithm to generate a list of prerequisites between the target concepts.

As done in [145], the task of PR candidates extraction from external knowledge graphs can be split in two steps: given a target concept C (i.e. a concept included in our text resource) and a knowledge graph $\mathscr{G}$, the goal is to (a) find candidate concepts in $\mathscr{G}$ that can be prerequisites of C; (b) evaluate or rank the prerequisite relations between C and candidates prerequisites using some algorithmic strategy. Wikipedia hyperlinks can serve to infer prerequisite between concepts in our document if we find a match between concepts and Wikipedia page titles (assuming a one-to-one correspondence between document titles and concepts). Such hyperlinks have also been used to predict whether reading a page in Wikipedia is a prerequisite of reading another page [214]. [68, 139] proposed a method to exploit Wikipedia for identifying pedagogical sequences of Learning Objects.

Relying on similar external graph resources is generally chosen because of their comprehensive set of knowledge items and their extensive relationships. For instance the hierarchical structure of a concept can be drawn from categories in DBpedia categorical system [145]. Wikipedia is also a valid resource since its pages are densely linked, and hence a document will likely be linked directly to any prerequisite page [214]. However, not all hyperlinks in a Wikipedia page will indicate a prerequisite [214] and, generally, external knowledge graphs might not have a page for a specific concept, or an existing page can be too short to provide a good coverage of its prerequisites [135].

2.3.1.3 Reference distance

The present subsection deals with an approach originally used relying on external knowledge, but that can be actually suitable also for approaches that exclusively focus on internal textual knowledge. Intuitively, if two concepts are bound by a prerequisite relation, then they are somehow close to each other. They can for instance share at least certain semantic aspects. Moreover, these two concepts awake our interest regarding their distribution inside a text. For instance, are they always co-occurring together? Do they share the same linguistic contexts? In computational linguistics and cognitive sciences, the fact that two terms co-occur in text corpora can provide a basis for semantic representations [131]. In particular, according to Distributional Semantics, we could infer semantic similarity between two linguistic expressions based on their distributional properties in large samples of linguistic data [131]. More in generally, similarity measures comprise a range of metrics largely used in Information Retrieval and text mining for detecting the closeness or distance between two items such as documents. Cosine and Jaccard similarity (or distance, i.e. their complement) are two examples of

such metrics.
A very intuitive yet robust link-based metric for prerequisite identification was proposed in [135], with the name of Reference Distance (RefD). As the name suggests, the focal point here is the notion of (co-) reference. This metric was conceived observing how citations between concepts work in texts, and it was partially inspired by cognitive semantics and frame semantics [135]. The linguist Charles Fillmore developed Frame semantics, a theory of meaning based on the notion of "frame", i.e. a structured and coherent set of related concepts [82]. According to this theory, we cannot really understand a concept without having access to the knowledge conveyed by all its related concepts. The assumption behind RefD is that for explaining some concept C usually we need to make a reference to at least some of its prerequisites contained in $\mathscr{P}$ (i.e. the set of all its prerequisite concepts), while for explaining some concept belonging to $\mathscr{P}$ normally we would not require to make frequently mention to the specific concept C. RefD has been originally employed in [135] using Wikipedia links between two concepts as reference relations among concepts. However, in our opinion this metric is particularly interesting for more than one reason:

- Compared to other intuitive or commonly used measures (such as co-occurrence, see later on, or cosine similarity), RefD better reflects some of the properties of the prerequisite relation, especially the asymmetry. Unlike metrics such as co-occurrence, in fact, RefD behaves as an asymetric co-reference, penalising pairs of concepts when both make roughly the same number of citations to the other. As a result, a direction can be assigned to the extracted PR following this counting.
- This co-reference might be intended as a hyperlink through items in an external resource (as it was originally implemented) but it could also be generalised and be intended as a co-occurence of terms in a window of textual context (i.e. an explicit reference).
- Lastly, as suggested by their inventors, RefD can be incorporated into existing supervised models as a significant feature, as did for instance in [154].

2.3.2 Approaches based on internal information

2.3.2.1 Lexico-syntactic patterns

Textbook authors can establish relations between concepts by means of cue phrases, and these linguistic expressions can act as triggers for identifying prerequisite rela-

tions. Since [108], pattern-based methods for hypernym-hyponym detection have been frequently used [108, 150, 185, 188]. Their essential idea is to perform pattern matching for finding in a text certain lexico-syntactic patterns that reveal a hypernymy relation, like for instance NP_y *is a* NP_x. As we can see from this pattern, for achieving better results, pattern matching is conducted on the POS-tagged text, i.e. tagged with parts of speech (NP here stands for noun phrase), and not just by performing string matching with regular expressions in the untagged text. A typical problem with patterns is that words must appear exactly with the predefined configuration, otherwise a relation will not be captured [186]. Moreover, text should be processed through an NLP pipeline (e.g., tokenized, POS-tagged and lemmatized) to avoid mismatches due to inflected word forms, and anaphora resolution as well as acronym expansion may be necessary to capture relations also when a word is replaced by a pronoun or when concepts are abbreviated. More sophisticated and complex rules are generally needed to capture many linguistic configurations, and even so they may not be able to identify all of them [188, 205]. These drawbacks led researchers to shift from pattern-based to methods exploiting distributional representations; however, some experimental evaluations show that pattern-based methods may be able to perform well with configurations that distributional representations still do not capture [186]. Hypernym extraction (whether or not pattern-based) is closely related to Definition extraction, i.e. the task of automatically identifying definitional sentences within texts [159]. As a matter of fact, the idea of extracting hypernymy relation for identifying prerequisites is based on the observation that concepts in textbooks are generally defined using some of their prerequisites. Thus, a concept definition is generally marked in the text by expressions such as "is a", “is defined as”, “is called”, “is known as” and so on, and these linguistic expression binds the two concepts. As also discussed in studies of terminology, many types of concept definitions typically make reference to one or more of its related (possibly prerequisite) concepts [190]. For instance, definitions by analysis provide the concept’s superordinate and also the specific attributes of the defined concept compared to other individuals of the same class (e.g. "pneumonia is an inflammatory condition of the lung") and definitions by synonym provide a generic term used in a not specific linguistic variety (e.g. "betula, commonly known as birch"). All that said, definition extraction is an open issue, since definitional sentences in real texts occur in highly variable syntactic structures [159]. Concepts can be for instance defined by periphrase (e.g. "icterus, i.e. the state when skin assumes a yellowish color"), by implication (i.e. giving an explicative context, e.g. "we make a divine fallacy when we say that something is paranormal just because is

incredible"), by extension ("programming languages are Java, Python, C++, ...") or even in mixed forms [190]. Also techniques for causal relation extraction can be based on pattern matching, besides machine learning algorithms exploiting annotated corpora (see [16] for a survey). Methods for causal relations extraction based on syntactic patterns have been proposed for instance in [27], that offers a model for identifying causal relations when they are marked (i.e. signaled by some cue phrase such as "because", "since") and explicit (i.e. linguistically expressed in the text). The limit of methods based exclusively on pattern matching is that causal relations are often instantiated in text without using syntactic markers, or even implicitly, i.e. through the semantics of the verbal constituents (e.g. the sentence "Louis XVI was guillotined in 1793" implicitly cause that "Louis was dead in 1794", even if the latter sentence do not appear in the text).

2.3.2.2 Textbook structure

Even if not enriched by any sort of external knowledge base, a textbook (or more in general a textual resource) still disposes its contents in a structured way. Text is usually organised in sections and paragraphs, and rudimentary mechanisms for hyperlink navigation are often provided also in paper books (e.g. table of contents, glossaries, index of terms). In many textbooks, relevant concepts are marked or highlighted in some manner, generally by using a particular formatting style such as bold or italic font. Often this formatting style is applied only when the concept is explained. In some books the concept is reported in side margins, in close proximity to definitional sentences. In case of web-based textual resources that are encoded in HTML it would be particularly easy to extract concepts embedded between tags defining such styles (e.g.): [53] extracted for instance these features for performing automatic resource sequencing based on prerequisite / outcome separation. But most of all, textbooks normally have a table of contents (TOC), which implicitly reveals some clues as to prerequisite dependencies between concepts across the sections. A reasonable assumption is that concepts are introduced in a textbook according to some order of precedence (e.g. from general to specific, or from basic to advanced). TOC and temporal order can then be taken into account to grasp prerequisite dependencies. In this respect, an early example of a simple algorithm for prerequisite/outcome separation was provided by [42]. This algorithm is based on the following assumptions:

1. While analyzing examples from some lecture, concepts corresponding to examples from all preceding lectures are considered to be completely learned.

2. In each example, all concepts introduced in the previous lectures are considered to be prerequisites to this concept, while the concepts first introduced in the current lecture as viewed as outcomes.

3. The set of new concepts found in all examples associated with the lecture is considered to be the pedagogical goal of the lecture

For its simplicity and common sense, the algorithm has been used as a valid baseline, as happened in [53]. Learning materials do not always reflect these intuitive assumptions though. For instance, the knowledge that appeared in previous units can be only partially required for the present unit. More recently, [236] proposed a distance metric (*TOC distance*) between two concepts, defined as the distance between their subchapter numbers, and the computed value was used along other features measuring complexity level difference between concepts. One issue of TOC based metrics is that we need to associate each concept with a section or set of sections where this concept is explained. Concept-section correspondences can be made with title matching (i.e. concept C is regarded as discussed in section S if it occurs in the title), but this will lead to a loss of concepts that are not presented at high levels of granularity. On the other hand, a simple occurrence inside a section may not be enough because the point where a concept is mentioned is not necessary the point where the concept is defined or explained (it can also be in fact only introduced). [3] proposed a method based on text structure and term relevance, where each concept is associated with the section in the textbook where it has the maximum relevance (as measured, for example, using TF*IDF).

2.3.2.3 Co-occurrence

Co-occurrence based methods are the core of many approaches for PR relation identification [99, 135]. However, while co-occurrence is an intuitive condition for PR, high co-occurrence is not necessarily a measure for PR strength, since it could identify also other types of relations, such as taxonomic relations, complex relations, general associations or co-requisites. Therefore, a reasonable assumption is that co-occurrence of two concepts is likely a necessary but not sufficient condition to identify a prerequisite relation. In general, in fact, high co-occurence frequency (i.e. counting how many times two concepts occur together in a certain span of sentences) is a good indicator of relatedness (as shown in previous works [60, 135]), thus it can also underpin other kinds of relations besides PR. The principle can be extended from the sentence level to a section level. Temporal order is the most natural criterion to give direction to the extracted relation

2.4 Prerequisite enriched datasets

The raise of interest in the use of Artificial Intelligence techniques for automatic prerequisite learning has favoured the development of annotated datasets with explicit labels for prerequisite relations. Such datasets are valuable resources for either training or testing machine learning algorithms against a gold standard representing the human expert behaviour. In the literature, the evaluation of a method for automatic prerequisite relations extraction is usually performed through comparison with a gold standard produced by human beings that annotate relations between concepts (see, among the others, [78, 135, 214]). Despite being time consuming, creating manually annotated datasets is an effective practice and produces gold resources, which are still rare. In fact, in the literature there is only a few datasets annotated with prerequisite relations between educational concepts. Most of such dataset consist of pairs of educational concepts enriched with a binary label expressing the presence or the absence of a prerequisite relation between the two concepts. Educational data used to build such pairs can be mainly of two distinct types of data:

- course materials, acquired from MOOCs [54, 94, 170, 171, 189] or university websites [133, 137];
- educational materials in a broader sense, such as scientific databases [98], learning objects [94, 154, 214] and textbooks [12, 143, 236].

Data from textbooks is the most rarely used, arguably because the goal of the resource creators was domain modeling rather than textbook modeling. In this work our goal is to model the content of a textbook, accounting also for the author's didactic preferences, hence we rely on the textbook only. As in our case (see 5.1), the most common approach for prerequisite annotation is asking annotators to evaluate all possible pairs generated from the combination of selected concepts [54, 133, 236] or a random sample of that set [94, 98, 171]. Another approach consists of letting annotators to autonomously create concept pairs based on their knowledge about the topic [143]. To the best of our knowledge, [214] is the only case where crowd–sourcing is employed for annotation: the authors infer prerequisite relationship between concepts by exploiting hyper-links in Wikipedia pages and use crowd-sourcing to validate those relations in order to have a gold training dataset for a classifier. Asking domain experts [78, 135, 136], or students [169, 234], to perform the annotation is the most frequently adopted approach, possibly justified by the fact that asking domain novices to perform the annotation is not beneficial, as said in

5.1 and reported in [12]. To the best of our knowledge, the only two non-English datasets annotated with prerequisite relations are the ITA-Prereq dataset in Italian [154] and two Chinese datasets [143, 247].

CHAPTER 3

ISSUES, RESEARCH GOALS AND METHODOLOGY

In the following section we will review literature concerning prerequisite relation and share with the reader questions that such literature let emerge.

3.1 Issues arising from the literature

Issues concerning prerequisite identification. As described in previous chapters, a growing body of literature has been focusing on automatic prerequisite identification. A question that may very easily comes to our mind as a natural starting point would be then: *how can we automatically extract prerequisites from textbooks?* Early works in this field relied on graph search algorithms [41, 226] or proposed intuitive rule-based algorithms that took into account the temporal order of concepts across the learning units [42]. More recently, efforts have been made towards the definition of link–based metrics using Wikipedia's hyperlinks between pages [135]. [214] made the first attempt to apply machine learning techniques to prerequisite prediction task: hyperlinks, hierarchical category structure and edits of Wikipedia pages were used as features for a Maximum Entropy classifier. Similarly, [94] use Wikipedia's hierarchical category structure and hyperlinks as features for a Multilayer Perceptron classifier. Differently from the above methods, [136, 138] also integrated text–based features along with graph–based features. In [141] the authors propose two approaches based on feature extraction and machine learning to map courses from different universities onto a space of concepts. Likewise, in [171] the authors define various features and train a classifier that can identify

PR relations from video transcripts. Both methods use semantics and context based features. [94] introduces a weak ontology driven approach: they extract lexical and semantic features and apply machine learning techniques for detecting a PR between learning objects. Machine learning approaches commonly have the disadvantage to require considerable amount of annotated data, which are not easily available in the case of PR identification, especially if our aim is textbook modeling (instead of domain modeling). Although several methods have been devised to extract prerequisite relations [135, 138, 170], they were mainly focused on educational materials already enriched with some sort of explicit relations, such as Wikipedia pages, MOOC materials or Learning Objects. Conversely, *a more challenging task is the identification of prerequisites when no such external relations are given*, and the textual content is therefore the only available source of information. The need to adopt a similar criterion of extraction arises from the observation that this would be: a) suitable for prerequisite learning also when external sources of structured information are not available; b) capable of inferring prerequisite relations directly from the educational material where concepts are described. Motivation b) represents a particularly desirable scenario, especially if we consider that a PR relation strictly depends on the writer's communicative intent and expository style.

Issues concerning evaluation approaches. A further issue concerning the automatic extraction is *how we can evaluate a new algorithm*. A common practice is comparing the algorithm performance with one or more metrics, called baselines, using a human annotated dataset as ground truth. Baselines are usually simpler and more naive than the new method, and the expectation is that a more sophisticated method should be at least able to outperform a baseline in order to be called successful. From the point of view of a researcher involved in PR automatic extraction, it can be very useful having a selection of extraction algorithms and baselines discussed in literature, so that they can be used to extract prerequisites from corpora or during the evaluation of a new method.

Issues concerning annotation. As already said, no matter which algorithmic strategy or baseline are adopted, *a labelled dataset is commonly required* when our goal is to develop an automatic method. Machine learning algorithms, in particular, heavily rely on annotated datasets for training, testing and evaluating their performance. Annotation or enrichment of textual datasets with prerequisite relations dates back to 2012, when [214] exploited crowdsourcing for this particular task. After that moment, most prerequisite-enriched datasets have been built by recruiting domain experts [78, 135, 136] or students

with a certain competence on the domain [169, 234]. Although expensive and time-consuming, the collection of manually labelled data from experts constitutes indeed the most recommendable practice for building reliable resources. A possible issue can be seen however when a prerequisite dataset is built by experts that rely on their own personal background knowledge. As a matter of fact, in similar cases a domain expert could make accidental or intentional use of his or her domain knowledge, therefore enrich the text with some relation that is ontologically legitimate but not really present in the text. The opposite risk also exists, i.e. the annotator, by relying on background knowledge, might overlook and fail to annotate a relation that is expressed in the text because it is not mirrored in his or her mental representation of the domain. Whereas such behavior might lead to enrich the text with a set of prerequisite relations that do hold from an ontological point of view, the problem arises when we are interested in studying how prerequisite relations are actually expressed and organised in educational materials. If this constitutes our main goal, a different type of annotation methodology is probably more useful, i.e. an annotation *strictly bound to the text that we are considering*. The prerequisite dependencies of a textbook should be traced in the really same textbook, since they might only partially overlap with ontological relations (such those encoded in a domain ontology), no matter how pedagogically debatable this discrepancy can be. In fact, even in the case of a poorly conceived learning material, that may be lacking of important dependencies or where the author explains concepts in an awkward order, it is reasonable to manually annotate (and consequently automatically extract) the relations that are expressed in the text. This choice can be motivated by the fact that a final user (e.g. a learner studying that textbook) will eventually cope with these relations and not with the ones that are included in a domain ontology or reflected in a particular expert's background knowledge. Generally, PR relations in real educational materials strictly depend on how the author chooses to present topics and concepts. As an example, consider top-down and bottom-up approaches, since both are largely used in textbooks: the former tends to explain a topic starting from broad concepts and definitions, while the latter starts from specific cases or examples. The influential educational psychologist Ausubel (see 2.1) advocated deductive teaching: general ideas must be taught first, followed by more specific notions [194]. This view assumes that learners' cognitive structures are hierarchically organized, so that inclusive concepts subsume subordinate ones. While this can be often true, inductive explanations are also commonly found in materials: authors can indeed first describe a particular case (because this is maybe more familiar to the learner) and then move up and reconstruct the general category or rule. A similar

line of though can be applied to procedural skills as well, whose description can be decomposed in sub-tasks and explained in more than one order (e.g. from the first step to the last, but also in reverse order). This is a common instructional strategy at least since the advent of Programmed Instruction (see 2.1.1), where two chaining procedures can be adopted (forward chaining, i.e. teaching from A to Z, and backward chaining, i.e. teaching from Z to A). Choosing one approach over the other obviously affects the direction of the PR relation in the text, i.e. from a general concept (or a initial step) to a specific concept (or a final step), or viceversa. Although the teaching of a subject may eventually come to experience a process of standardization, the order of contents to be presented within a course is still largely a matter of the author's preference. In foreign and second language teaching, for instance, the sequence of acquisition is an ascertained notion used for describing a possibly fixed and universal order in which all learners tend to acquire grammatical features of the target language. Steven Krashen, in particular, asserted that all students acquire the language roughly in the same order (natural order hypothesis) and the best that a teacher can do is providing their student with a comprehensible input defined as $i + 1$, i.e. the very next item along the sequence of acquisition after their current intake i [123]. These lines of research were so influential that most foreign language textbooks tend to present grammar topics and rules in a predictable order, also because well established frameworks for evaluating linguistic competence suggest to follow similar sequences (e.g. Common European Framework of Reference for Languages). However, this predictable and "natural" order cannot be easily found in every subjects. Let us take a look, for instance, at the field of Programming Languages. Although there could be some shared practices suggested by our common sense, the order here largely varies depending on the textbook. As a result, take two classic and appreciated programming language books, one for C [117] and the other for C++ [211]: the former explains *while loops* first and then *for loops* (because *for loops* can be rewritten if you know *while loops*); in the second case, *for loops* are explained before (as a more general iteration statement), then *while loops* (as a specific case). The domain could also affect PR identification in a more general way. Subjects such as Computer Science, Algebra and Physics are conceptually heavy and besides we can expect that here prerequisite relations between concepts are clearly expressed in the text. On the other hand, in some disciplines belonging to the Humanities, such as Arts, Philosophy, and Literature, the problem could be much harder to tackle because of the fuzziness of concepts and relations and the stylistic contortions that the author may weave.

Issues concerning inter-annotator agreement. Among other issues involved in the computational treatment of PR relation we can mention the inter annotator agreement [54, 78, 99], which refers to how differently two or more people agree on the annotation of the same text. When annotation is performed by non–experts, agreement is usually very low, thus an expert may be consulted to validate or revise the data [54, 99]. Such a low value can be arguably due to the fact that non–experts may experience difficulty when reading a text in an unknown or not-fully-mastered domain, therefore their opinions tend to diverge because of a lack of good understanding of the text. Our experience ([12]) and the literature [78] show that human judgments about prerequisite identification can considerably vary, even when the annotation is performed by experts and a clear explanation of the task is also provided. This phenomenon can depend on several factors, including the subjectivity of annotators and the type and complexity of the document. *Low agreement values also suggest that prerequisite relations in educational texts are commonly instantiated in an implicit form*, i.e. the author rarely announces their presence with a clear statement such as "this concept is a prerequisite of this other concept". Dependencies are rather expressed in a more or less ambiguous manner within the flow of the discourse, possibly but not necessarily triggered by some textual or visual cue (e.g. lexical patterns, formatting styles, see 2.3). Given this nature, building chains of concepts based on their pedagogical dependencies is not an easy task, either because it is hard to disambiguate between prerequisite and outcomes or because there is often a blurry boundary between prerequisite relations and other types of relations. For some annotation tasks (e.g. POS labelling, image classification, etc.) there is a well codified procedure, thus ambiguity, though still present, is limited to a subset of hard-to-annotate cases. On the contrary, for some other task ambiguity should be rather considered an inherent property of the task itself: prerequisite annotation, for instance, is intrinsically ambiguous per se. The way how an author explains concepts can range from a straightforward style (e.g. a sequence of sharp and concise concept definitions) to a more convoluted stream of discourse. In Problem Based Learning materials and tasks are frequently presented in such a way that a learner first faces an advanced topic/concept or skill (for which she still has not encountered or mastered the prerequisite knowledge), and then she is encouraged to discover by herself the prior knowledge she should own to overcome the problem. Spiral Curriculum based learning paths usually present the same topics several times within a sequence of learning units but from different points of view or at increasing levels of complexity. This also poses to the annotators the problem to deal with different levels of granularity. In general, evaluating the inter annotators'

agreement can thus be useful to assess if the text was intrinsically hard to annotate or whether the gold dataset can be trusted or further annotators are instead required.

Prerequisite annotation suffers from the fact that it is hard to properly define the problem and this might affect the building of appropriate and reusable datasets. Lack of guidelines, different annotation schemas or different ways to encode prerequisite (see 2.2) brings to datasets that are not easily comparable. For this reason, regardless the annotation methodology or the inter-annotator agreement, annotations should also be presented with *a systematic description of the guidelines provided to annotators*. For instance, we observe that in the mentioned related works prerequisite relation properties (i.e. irreflexivity, anti–symmetry, etc., see 2.2.1) are rarely taken into account in the annotation instructions for annotators. For example, the fact that a concept cannot be annotated as prerequisite of itself is usually left unspecified. Similarly, attention should be also paid to the fact that prerequisites are transitive relations, therefore if $A \prec B$ and $B \prec C$, then $A \prec C$.

Issues concerning PR annotation tool. Speaking about the practical procedure itself, most of annotation tasks consist in enriching a text with an explicit information about some of its features, eventually generating a machine readable format (like for instance a tabular file, or a text encoded with some markup language such as XML), in order to be later automatically processed for analysis. In similar cases, the goal of annotation tools (see [160] for a recent survey) is to support the creation of better datasets, for modelling a certain phenomenon (in our case prerequisites in textbooks) and allowing to train algorithms. To support the annotation of prerequisites between pairs of concepts, [99] developed an interface showing, for each concept of the domain, the list of relevant terms and documents. Although this can be of some support for the annotation providing certain useful information, it cannot be considered an annotation tool itself. According to our knowledge, a tool specifically designed for prerequisite structure annotation which also features agreement metrics is still missing.

For instance BRAT [208], one of the most popular annotation tools, supports the annotation between spans of text such as dependency structures. Even if BRAT is highly flexible, it does not provide the following feature of our interest: *(i)* connecting spans of text does not cover all the possible cases of PRs that we want to identify, since PRs can also occur between terms not sharing the same textual context; *(ii)* since the annotation of PR is an ambiguous task, an adjudication interface (i.e. an interface that allows a final expert to easily revise and merge different annotations [83]) is required for

combining annotations, computing agreement between them and eventually creating gold standards; *(iii)* lastly, loading of pre-selected domain concepts is another desirable feature. The *(ii)* requirement is fulfilled by WebAnno [76], which is a web-based and visually supported system for distributed annotation with a wide range of linguistic annotations including various layers of morphological, syntactical and semantic annotations and some automatic annotation suggestions to increase the annotation efficiency. TagTog [51] is a web based versatile collaborative annotation tool for entities and relations which is particularly suitable for annotating large texts since it leverages manual user annotation in combination with automatic machine-learned annotation via pre-trained models. The tool does not resolve the *(i)*, while it supports the human revision phase *(ii)* and the import of a dictionary for the *(iii)*. The aforementioned tools can be opposed to famous crowd-sourcing platforms (e.g. Amazon Mechanical Turk, Figure Eight), usually employed when the annotation task is guided by simple rules or common sense.

Issues concerning PR analysis and visualisation. Beside being useful for machine learning algorithms, textual annotation produces as an outcome language resources, which are fundamental in Corpus Linguistics and NLP since they can be used not only to develop and evaluate new systems but also to capture and model linguistic phenomena and therefore perform analysis on them. Datasets can be designed not only to train ML systems but also to get a thorough understanding of the prerequisite relation in real educational data by looking at how this phenomenon is instantiated in texts. In the case of PR this is even more important because *as a semantic relation, prerequisite relation has not been well studied in computational linguistics* [135].

As a branch that seeks to marry analysis processes with visualisation tools, Visual Analytics aims to handle large amounts of multidimensional data by means of interactive graphic interfaces and advanced visual representation techniques during the process of analysis [115].

The effective integration of visualisation technologies in curricula with the purpose of facilitating teaching and learning of abstract concepts has been already investigated (see for instance [158] for a study on visualisation and learner's engagement in Computer Science education). More recently, in Educational Data Mining (see [187] for a survey), several studies are oriented toward visualising different kinds of educational data. In the field of Learning Analytics, information visualisation techniques have been studied to empower learning dashboards with graphical representations of the learning process [75]. To the best of our knowledge, *a specific contribution on how information visuali-*

sation techniques can be applied to the analysis of prerequisite relations in textbooks is still missing in the literature. [151] employed two representations (Hierarchical Edge Bundling and Hive Plots) on the structure of a book to show how these visualisations can deal with large graphs that have a hierarchical nature. However, he did not further develop the investigation on textbook prerequisites by means of visualisation.

3.2 Research goals

In this section we highlight the main research goals and questions of the thesis. They emerge from the issues discusses in the previous section and will guide to the design of the PRET framework.

- Our first goal is to define a methodology for systematically annotating prerequisite relations in textbooks. This goal is functional for analysing the PR phenomenon and for evaluating and training automatic methods of extraction. It addresses the research issues that were discussed in the previous sections and that can be summarised as follows:

  (1) Which annotation schema should we use for the annotation? (e.g. a full graph at sentence/paragraph level, a prerequisite / outcome binary classification at section level, etc.)

  (2) What kind of knowledge the experts should include in their annotation? (i.e. should they try to capture only the information present in the text or use instead also their background knowledge?)

  (3) Can we define clear guidelines for annotators?

  (4) Which tool can we use for the annotation?

  (5) How to deal with inter-annotator agreement?

  (6) How can we merge different annotations and create a gold standard?

- The second goal concerns the automatic extraction of prerequisite relations in textbooks. This goal addresses the research issues discussed in the previous section, and are summarized as follows:

  (7) Which approach for automatic extraction (i.e. external knowledge or internal knowledge?)

(8) how to evaluate with respect to the peculiarities of prerequisite relation?

- Both goals require analysis :

(9) Which kind of analysis can be performed on annotated data?

(10) Which visualisation techniques can be used for this particular problem?

We will now try to group these issues together according to a sensible criterion. Issues from 1 to 4 regard the fundamental moment of manual *annotation* (annotation schema and methodology, annotation guidelines, annotation tool). Issues 5 and 6 arise once we have collected different annotations and we want to make comparisons between experts opinions and *combine* such distinct annotations in a gold resource. Issue 9 arises when a researcher wants to draw information of various kind (e.g. quantitative or linguistic) from data through *analyses*, while issue 10 refers to the task of data *visualisation* (either annotated or extracted relations). Finally, 7 and 8 strictly concerns algorithms for prerequisite automatic *extraction* and *evaluation*. Since all these represent different but inter-related tasks in a larger process, we propose that they should be consistently addressed within a comprehensive theoretical framework comprising the following components:

- Annotation (1-4)
- Combination (5-6)
- Extraction (7-8)
- Analysis (9)
- Visualisation (10)

The name we gave to this framework is PRET (Prerequisite Enriched Terminology), since it finally produces for a given text a terminology enriched by its prerequisite relations; this can then be used as a dataset for different uses or in educational applications for augmenting the textbook with an explicit representation of its learning dependencies. The framework will be described in details in the next chapter, while chapter 5 will be dedicated to its development and its applications.

3.3 Contribution of the thesis

In the present thesis we make the following contributions:

1. **Framework design and development.** To the best of our knowledge, this is the first comprehensive framework for annotating PR relations that also supports researchers in all the steps of PR structuring from textbooks. As discussed in the previous section, other tools can support some of the phases but not the complete flow, while in the framework we present here annotation is integrated with the other phases.

2. **Resources' multipurpose-use.** The result of the annotation, i.e. the Prerequisite-Enriched Terminology, is versatile as it can be helpful in several tasks. For example, it can be used as a dataset to train machine learning algorithms, as gold standard for the evaluation of PR extraction algorithms, or to generate dynamic learning paths and to enrich textbooks.

3. **PR annotation in context.** As a linguistic annotated resource, a Prerequisite-Enriched Terminology constitutes a precious instrument for studying PR as a linguistic phenomenon. One of the major advantages and contributions of the framework is indeed to encourage a text-driven and context-anchored annotation, i.e. experts annotate relations actually conveyed by the text itself (not by other sources, including their background knowledge) and the inserted relations are anchored in the linguistic context where they take place according to experts opinion. This allows to investigate relevant contexts at different levels of linguistic analysis, possibly gaining a deeper understanding about how the phenomenon is instantiated in texts and eventually confirming or discovering insights regarding the linguistic features we should take into account for the automatic identification of prerequisites. In this sense, we provide in 5.2 examples of how we can conduct such sort of linguistic analysis, in conjunction with other forms (graph analysis, quantitative analysis).

4. **Annotation schema and methodology.** We defined an annotation schema that enables the annotation of PR relations in the linguistic context where they appear. Moreover we defined a PR code book (appendix A) and PR annotation guidelines (appendix B) expanded with suggestions to support experts (e.g. eliciting questions, also in appendix B).

5. **Text-based extraction.** Consistently with our idea that PR relation is largely affected by the characteristics of the text where it is found, we propose extraction approaches that, contrary to many others, aim to extract PR from unstructured text, i.e. without exploiting external structured knowledge. In particular, in 5.4 we present a new method based on burst analysis, co-occurrence, and temporal reasoning. Similarly, in section 5.4.3.2 we describe a deep learning based approach that exploits textual features for extracting prerequisite relation between educational concepts in a textbook.

6. **PR Visualisation.** In section 5.5 we present a collection of visualisation techniques conceived to help researchers in their effort to better understand issues related to prerequisite dependencies in textbooks and developing more powerful strategies for the automatic extraction. To the best of our knowledge this represent the first contribution that is specifically addressed to apply visualisation techniques to the PR research problem.

7. **Annotated resource.** We produced a gold dataset manually annotated with prerequisite relations between pairs of concepts occurring in a text. Even if limited in its dimension, this dataset is available for the community and for further investigation on the PR phenomenon.

We can see that these contributions addresses the issues 1-10 listed above. In particular, contribution 3 addresses issue 9, contribution 4 addresses issues 1-6, contribution 5 addresses issues 7 and 8, contribution 6 issue 10.

CHAPTER 4

PRET FRAMEWORK

As anticipated in the previous chapter, in order to support researchers in their efforts to deal with issues involved with prerequisite relations in textbooks, we designed a framework called PRET [1]. The first step for the definition of this framework was the development of the annotation module as a standalone tool for supporting the process of annotation of prerequisite relations. Such tool is the result of our experience in the process of manual PR annotation and analysis related to issues of PR identification and automatic extraction: during an initial study on PR manual annotation (described in 5.1.1 and whose reports were firstly reported in [12]), we observed several difficulties encountered by experts when addressing the task. We will discuss such issues in the next chapter. To investigate such difficulties, we expanded our research as described in the previous chapters. The PRET framework has been designed to address the research questions resulted from such investigation and reported in 3.2. The aim is to support and facilitate experts during all phases associated with the structuring of PR relations: from textual pre-processing to data visualisation, including manual annotation, data analysis and extraction. In the following sections we will describe the architecture of this multi-module framework (shown in 4.1).

[1] Prototype is available at https://github.com/Teldh/PRET (a screenshot is shown in fig. 4.2).

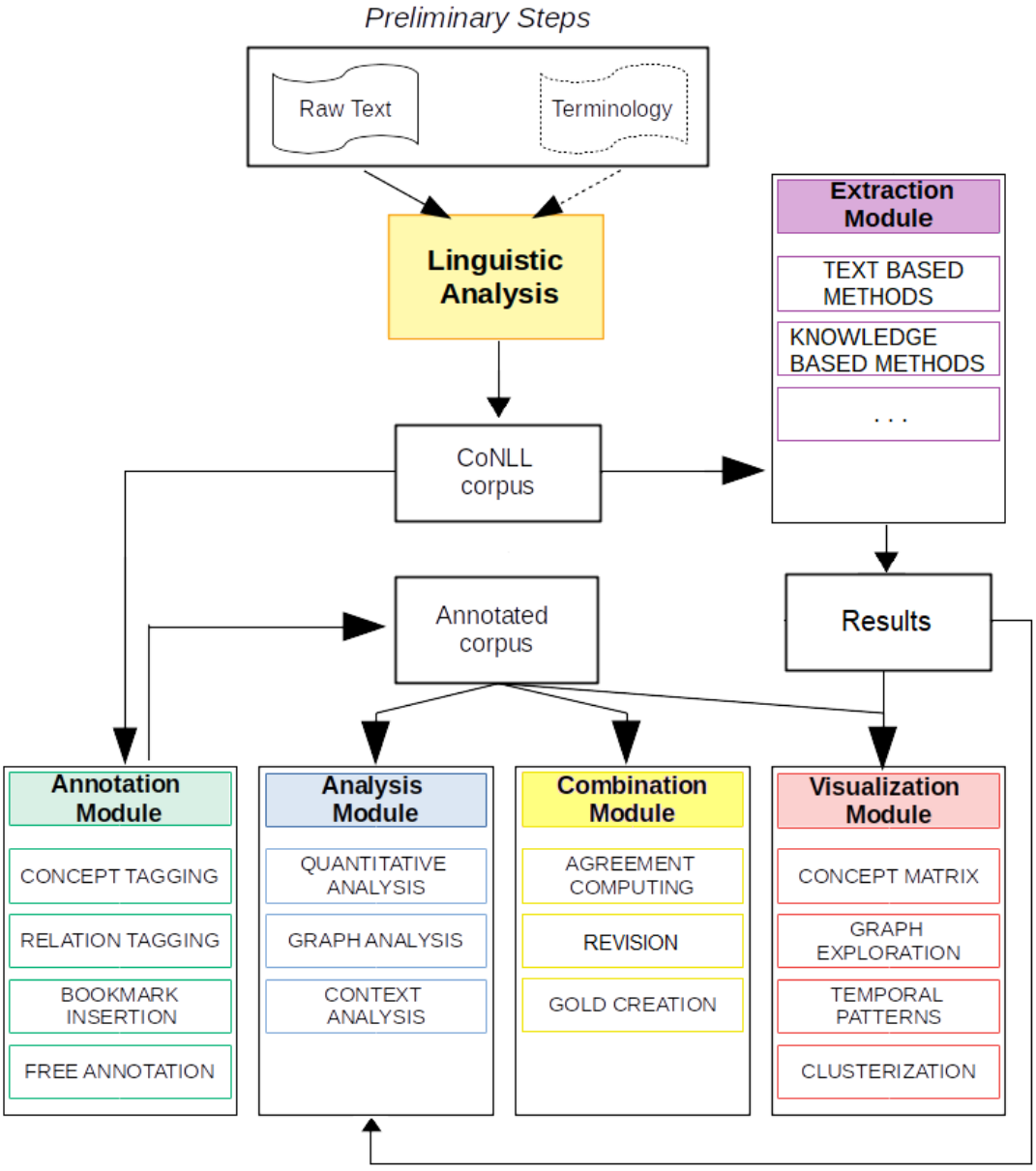

Figure 4.1: Framework architecture.

4.1 PRET Architecture

Figure 4.1 shows the main components of PRET architecture. As shown in figure, the initial workflow supported by PRET includes a phase of pre-processing. Before performing any task on a new text, as a first step the user uploads a textual corpus and an optional terminology representing domain terms extracted from the corpus itself. The optional terminology can be useful in cases when annotators need to work with a shared set of

ID	FORM	LEMMA	UPOS	XPOS	FEATS	HEAD	DEPREL	DEPS	MISC

Table 4.1: Fields in CoNLL format

.

concepts, automatically extracted or manually provided by an expert.

The input text is processed through the linguistic analysis pipeline UDPipe [210] that provides linguistic annotation at various levels, from sentence splitting to part-of-speech tagging. The output of the analysis is returned in CoNLL format[2] in order to comply with the standard format for linguistically annotated texts, so that a text pre-processed with a different tool can be also used in PRET. CoNLL is a format that is used in several shared tasks in NLP [43]; it basically consists of tabular data contained in a plain text file where all the linguistic information extracted from a corpus is encoded. More in details, each token of the corpus is represented in a CoNLL file as one row with tab separated values, while a blank row marks the end of a sentence and the beginning of the next one. All the information regarding a particular token is reported in the fields shown in Fig. 4.1: identification number of the token inside the sentence (ID), the linguistic form which actually occurs in the corpus (FORM), the lemmatized form (LEMMA), the universal (i.e. core) part-of-speech tag (UPOS), the language-specific part-of-speech tag (XPOS), morphological features (FEATS), information on syntactic parsing according to a dependency grammar representation (HEAD, DEPREL, DEPS) and finally additional information (MISC).

In our view, having a linguistically annotated text is fundamental since it allows us to disambiguate concepts based on their part-of-speech and their normalised base form (i.e. lemma). Moreover, and most importantly, the linguistic analysis offers the possibility not only to extract all sentences where concepts and their prerequisites occur, but also to identify all the linguistic structures underlying the relations. This allows, in phases of analysis, to investigate annotations from a linguistic point of view and at fine-grained level.

The second part of the pre-processing phase is subordinate to the presence of a terminology, since it consists of finding all terms in the parsed text. This is thought to help the annotator to notice the presence of a pre-selected concept while reading and maintain his/her attention at a high level. All the concept instances are annotated in the CoNLL file according to the "IOB" tagging scheme (i.e."Inside–Outside–Beginning") originally presented in [181] and widely used in text chunking tasks such as Named

[2] `http://universaldependencies.org/format.html`

token	IOB	Concept ID
a	O	_
metropolitan	B	12
area	I	12
network	I	12
is	O	_
a	O	_
network	B	1
of	O	_
intermediate	O	_
size	O	_

Table 4.2: Example of OIB-tagged educational text.

Entity Recognition. According to this schema, every token in a text is marked with "B" if it is the first token of an entity (such as, in our case, an educational concept), "I" if it is an internal or also the last token of an entity, "O" if it is not part of any entity. For example, the sentence *A metropolitan area network is a network of intermediate size* would be represented as shown in table 4.2, making clear that two concepts exist in the sentence, namely a multiword concept (*metropolitan area network*) and a single-word concept (*network*), both associated with their unique ID, which in turns is associated with every other occurrence of the same concept (even when it appears with a different form, such as the plural *metropolitan area networks*).

Once the pre-processing is completed, the expert can proceed to the manual annotation of PRs. Core modules of PRET support both annotation and its analysis, together with the creation of gold datasets (in cases of multiple annotators on the same text) and the extraction. In what follows, we will present each module and its features.

4.2 Description of the core modules

4.2.1 Annotation Module

The annotation module is probably the most fundamental since it supports the process of manually annotating the input corpus with PRs between pairs of concepts.

As said, prerequisite relation indicates what information one has to study/ know first in order to understand a given topic, hence in guidelines for the annotators (see appendix B) we defined PRs as binary relations between a target concept and a prerequisite concept, both belonging to the terminology of the same corpus. The annotation module is

conceived to facilitate the creation of concept pairs while reading the text: if the annotator believes that, in order to understand a target concept he/she is reading about in the text, it is required to master the knowledge related to one or more concepts, he/she can enter one or more PRs to represent this condition by choosing the prerequisite concept(s) from the terminology. Once the concept pair is created, the annotator can also define the weight of the relation, i.e. either weak or strong, depending on how much the prerequisite concept is relevant in order to understand the target one. The main advantage of creating PRs while reading a textbook is that each PR can be associated with the textual context of the target concepts that triggered the relation through a unique context ID which gives the concept coordinates in the text (i.e. the sentence where it occurs). As a consequence, the same PR can be added multiple times in different part of the text and associated with different weights depending on how the relation is presented in the specific context. In summary, context ID, prerequisite concept C_1, target concept C_2 and relation weight w are the information encoded in an inserted relation:

$$PR = \langle ID, C_1, C_2, w \rangle$$

As a result of the aforementioned procedure, the annotation does not produce manually created non-PR pairs: they remain implicit in the annotation and can be obtained by automatically creating all missing pairs of concepts as negative pairs (non-PRs).

In addition to creating PR pairs, the tool also supports the user in inserting new terms with the aim of either creating from scratch or enlarging a terminology (if it was imported) during the annotation process. This is done to make the process of modelling textbook content as natural as possible: some underestimated concepts (i.e. not present in the original terminology, either because not automatically extracted or because not considered relevant by the experts) might seem relevant to the annotator when reading the text, thus we offer the possibility to add them to the list of concepts. Once the concept is added to the terminology, it can be used to create PR relations by pairing concepts. Considering that adding a new concept to the terminology results from the annotator's reasoning about the content he/she is reading and could have cascade effects on the annotation (e.g. it may involve adding other concepts or creating PRs otherwise not possible), the list of manually inserted concepts is personal for each annotator. As a minor functionality, the module also allows the user to insert bookmarks for labelling significant sentences and add free textual comments, such as descriptions for those sentences. Moreover, since the annotation usually requires a substantial amount of time, a user can save his/her work and resume it later.

4.2.2 Analysis Module

As explained in section 3, annotation produces linguistic resources, which can be then analysed by experts to gain a better understanding of the problem. This is useful especially in case of poor studied phenomena as prerequisite relations. Taking as input the annotations resulting from the previous module, the Analysis Module produces results that provide an overview of the annotation characteristics and, if more than one annotation is available, similarities and differences between pairs of annotations.

The module performs the following type of analyses:

- **Quantitative analysis.** It reports basic and descriptive information about the annotation, such as the numbers of inserted PR quadruples, either weak or strong, and how many new concepts were added to terminology by the annotator.

- **Graph analysis.** The manual annotation can be represented in form of a direct graph, where concepts are nodes and edges represent PRs, hence we can perform network analysis to describe the characteristics of the resulting graph. In particular, considering the main properties of the prerequisite relation, we perform analysis on transitivity, connectivity, loops, average in-degree and out-degree, disconnected nodes, diameter of the network, longest path and number of source nodes and sink nodes.

- **Linguistic analysis.** It allows one to retrieve the textual context of an inserted PR and investigate its linguistic features. This analysis is possible since during the annotation process the annotator identifies relations and anchors them to the specific sentences in the text where these relation are established. The linguistic analysis comes with an interface called *Prerequisite In Context* (PIC)that allows the user to query the annotated corpus and retrieve all the relations that match the querying criteria. For each of these results the user can read the sentences, analyse POS and lemmas of the tokens and explore a graphic representation of the dependency parsing.

4.2.3 Combination Module

This module is designed to combine the annotations in order to obtain a gold standard dataset. Moreover the module includes the support for gold revision and agreement computation between annotations performed by different raters.

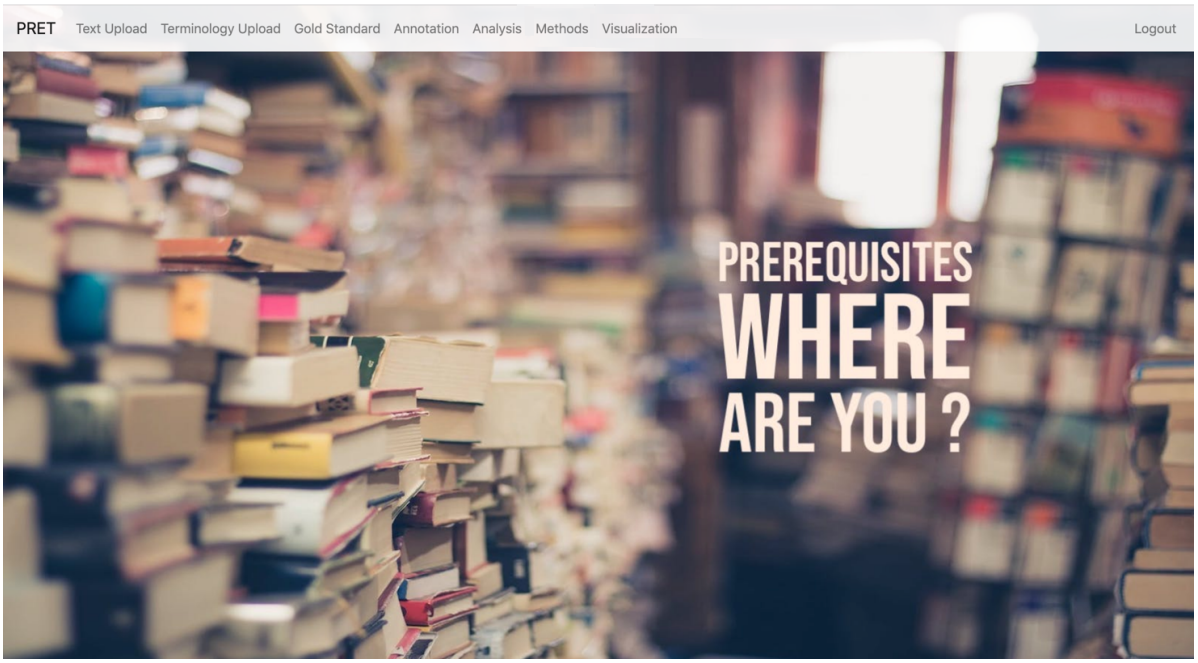

Figure 4.2: PRET prototype main page

- **Agreement computing.** This set of functionalities involves computing metrics of agreement between two given annotations. Inter-rater agreement metrics are widely used for capturing how many items in an annotated corpus received the same label by different annotators. A certain degree of agreement can be easily described as a percentage over the total number of items (raw agreement). Although in many cases raw agreement captures quite well the dataset characteristics and identifies potentially problematic areas in the annotation, more sophisticated coefficients are available in the literature. These measures are based on the assumption that if different raters produce consistently similar results, then we can infer that they have internalised a similar understanding of the annotation guidelines and we can expect them to perform consistently under this understanding [15]. The Cohen's k coefficient [62] and its variants are the most widely adopted agreement metrics, especially for linguistic and semantic annotations. As has been noted, k can be affected by skewed distributions of categories, a condition known as the prevalence problem, and by the degree to which the coders disagree, i.e. the bias problem, [77]. The first condition typically causes k scores to be unrepresentatively low, while the second condition typically causes k scores to be unrepresentatively high. [105]. Despite these potential effects, metrics belonging to the k family are as a matter

of fact usually considered more robust than raw agreement, since they take into account the possibility of agreement occurring by chance, i.e. agreement expected between the two raters if their classification was made randomly. The inter-rater agreement metric offered in PRET are *1)* Cohen's k [62] and *2)* F1 score. Using F1 score to evaluate the similarity between two annotations is recommended when one of the two datasets of the comparison is a gold standard. Since the annotation could involve more than two annotators, we also provide the Fleiss' extension of k [84]. Previous work addressed the task of PR annotation between pairs of concepts as a labelling annotation task using a small set of predefined categories, e.g. given a pair of concepts, decide whether or not they show a prerequisite relation [54, 171, 236]. This approach allows the straightforward use of popular agreement measures such as k, which is generally reported for prerequisite enriched datasets, so we decided to implement it in PRET in order to allow the user to compare his/her dataset with others.

- **Annotation Revision.** Experts can make errors during annotation, because of distraction or misunderstanding of the annotation schema. Furthermore, merging two or more individual annotations may give rise to conflicts, such as cycles or symmetric relations, that affect the consistence of the final gold standard and leave the experts with the issue of dealing with them. For this reason, the tool provides assistance during the revision phase that takes place after annotation or when combining different annotations. During the annotation phase, the tool can automatically detect candidate errors such as symmetric relations. After annotation, thanks to the context-anchored methodology of annotation, an expert can more easily review relations and decide how to handle them according to the context. In prerequisite relations this merging and revision phase is particularly difficult to perform in automatic way because, similarly to what happens in temporal relation annotation merging [142], we do not know a priori which element in each individual annotation is correct.

- **Gold Creation.** Gold standard datasets result from the combination of two or more manually produced annotations. This process is carried out according to one of the following criteria: *i)* include the relations inserted by at least one annotator; *ii)* include only the relations inserted by all annotators; *iii)* include only the relations inserted by at least half of the annotators; *iv)* include all relations created by at least one annotator and assign a weight reflecting the number of annotators that

inserted the PR, i.e. the higher the weight, the higher the number of annotators that added the relation (e.g. three annotators created the PR pair A,B, thus A,B will be part of the gold dataset with a weight equal to 3). Arguably, the goal of a particular study should guide the way one combines annotations. As said, the annotation task is designed to create a valuable resource for analysing phenomena and training systems. Criteria *ii)* and *iii)* represent a sort of majority voting, a standard consolidation procedure for labelling annotation. Criteria *i)* and *iv)* are less standard practices for gold standard creation but their use can be justified by the type of analysis to be carried out.

4.2.4 Extraction Module

This module provides a selection of methods for the automatic extraction of PR that can be launched on user's demand. The selection includes methods discussed in literature and exploited for PR extraction, but it is potentially open to new insertions. PRET currently includes the following methods:

- **Semantic relations**. As seen in 2.3.1.1, hypernyms-hyponyms and holonyms-meronyms can partially reveal PR. These relations can be easily determined using Wordnet, and APIs for accessing this lexical database through many programming languages are largely available.

- **Lexico-syntactic patterns**. As seen in 2.3.2.1, PR can be also extracted directly from texts using pattern matching for finding instantiations of hypernyms and taxonomic relations, as well as definitional sentences. In particular, we propose to use the selection of patterns presented in [236] since they were specifically conceived to detect PR rather than other relations.

- **Co-occurrence and temporal order**. As mentioned in 2.3.2.3, PR tends to be associated with co-occurrence. Implementing a co-occurrence based PR method or baseline implies making some choices regarding: a) the criterion to follow to define co-occurrence, and b) the direction of the extracted relation. As for a), we propose to take into account a window of context with a three sentences span width. If s is the index of the sentence where some concept C_1 occurs, then $s \pm 1$ is the window of context where we must look for another, hopefully PR-related, concept C_2 (i.e. the previous, or left, sentence, and the next, or right, sentence).

- **Textbook Structure**. Since our goals is extracting prerequisite for doing textbook modeling, we cannot neglect the importance of textbook structure, which reveals implicit information on the pedagogical dependencies between explained concepts (see 2.3.2.2). For this framework we opt for the metric defined in [236], called *TOC Distance*.

- **Concepts co-reference**. As seen in 2.3.1.3, the way how two concepts refer to each other can tell us a lot about their prerequisite dependencies. In particular, we recommend to use the RefD metric proposed in [135], for its intuitive meaning and its reported ability to perform well on medium-large corpora.

- **External Knowledge Graphs**. As seen in 2.3.1.2, external resources such as Wikipedia and DBpedia can be queried for finding in their knowledge graphs candidate prerequisites that may also be reflected in our textbook. In this framework we propose to rely on the Wikipedia-based approach presented in [235]. This assumes that each title of a domain related Wikipedia page is a concept; then it utilises three sets of features to infer a pedagogical hierarchy between a given pair of concepts A and B:

  - Usage in definition. As we said in 2.3.2.1, definitions often convey prerequisite relations: if concept A is used in B's definition, then A is likely to be B's prerequisite. For extracting definitions from Wikipedia pages [235] assumes that the first sentence in each page is a definitional sentence for the concept described in the page. This sounds reasonable, since encyclopedias or similar reference works (e.g. dictionaries, compendia, etc.) normally start their entries with a definition.

  - Content Similarity. If two pages cover similar topics, it is likely that the two represented concepts have some learning dependencies, i.e. either $A \prec B$ or $B \prec A$. Lexical similarity between Wikipedia pages can be measured with cosine similarity, as done in [235]. This assumption, however, is more critical than the first one, for two reasons: content similarity i) is a necessary but non sufficient condition for PR (in fact not all pairs of pages with similar content have a prerequisite relation), and ii) does not tell us the direction of the relation (if this exists). For these reasons, [235] proposes to identify pages that may be covering similar topics but are not at the same level of learning, as explained below.

  - Learning Level. Concepts with a lower learning level should be indeed learned first. According to [234], such level can be inferred with two features:
    * range of topic coverage: the more topics that a concept covers, the more basic the concept is. To do so, [235] run a topic model on the collection of Wikipedia pages describing the concepts of our text in order to generate topic distributions for each concept.
    * number of in-links and out-links: considering the graph nature of Wikipedia, cross-page links between its pages can be useful for detecting concept learning levels. According to [235], If A receives a lot of in-links from other concepts, it is likely that A is fundamental among all the concepts and thus should be learned first (a similar conclusion can be drawn when counting the number of out-links of a page).

- **Concept hierarchy**. External knowledge bases such as Wikipedia can also be used to perform automatic extraction of concept hierarchies from textbooks, as described for instance in [234]. Similarly to the previous method, candidate concepts in Wikipedia can be found by means of title matching between textbook and Wikipedia page titles, as well as by computing cosine similarity between lexical content of chapters and Wikipedia entries. Learning order can then be inferred using two heuristics:

  - definition in the first sentence (as in the previous method);
  - TOC (if the Wikipedia pages under examination have it): given two Wikipedia concepts w_i and w_j and their TOC toc_i and toc_j , w_i is the prerequisite of w_j if w_j appears in toc_i.

  Compared to the previous method, this leverage less graph hyperlinks between pages and tries to apply the TOC-based ordering criterion to Wikipedia instead of directly to the textbook. However, in Wikipedia TOC may be problematic, since not every page has it, and two Wikipedia concepts can appear in each other's TOC, making difficult to figure out which concept should be learned first.

- **Burst analysis**. Finally, a novel method is also added, called burst analysis for prerequisite extraction, which will be discussed in further details in 5.4.

4.2.5 Visualisation Module

The present module allows to visually explore the annotation by means of several dynamic and interactive graphic representations. Starting from [174], we tested a range of visualisation techniques and then we selected those that we consider as the most appropriate to visualise educational relations in texts. Here we describe only such techniques.

- **Concept Matrix Visualisation**. This is a dynamic and interactive representation of a $|T| \times |T|$ asymmetric adjacency matrix M, where each coloured cell $M_{i,j}$ represents a prerequisite relation between concepts i and j (see **Fig. 5.13**). Different shades of the same colour are used to encode different degrees of inter-agreement among annotators. The matrix arrangement is dynamic, i.e. the concepts along the matrix can be sorted according to different criteria: order of first appearance in the text, alphabetical order, frequency and cluster membership. See 5.5 for results obtained using this visualisation.

- **Graph Exploration**. Several variants of network-like representations (see for instance 5.12) can be used during the exploration of the annotation to visually detect elements such as loops (as resulting from human errors during the process of annotation) and transitive edges. However, as the dataset becomes larger, a concept graph becomes harder to explore if no filtering functions are implemented. For this reason the module also provide graph visualisations that allow decomposition into sub-graphs belonging to individual annotators, helping the analyst to investigate how annotators with a different profile produce different annotations. See 5.5 for results obtained using this visualisation.

- **Temporal Patterns Visualisation.** The main purpose of this sub-module is facilitating the analysis of temporal patterns established by the concepts along the flow of the text. Our interest in applying temporal analysis largely arises from the temporal nature of the PR relation. This intuition has also led us to develop a method for PR automatic extraction based on burst analysis, co-occurrence and temporal reasoning [2]. "Bursts" are the intervals of sentences covered by a concept where this concept is particularly relevant. With this sub-module we can visually analyse temporal patterns using a Gantt diagram that shows the bursts of concepts along the horizontal temporal axis (time can be measured in sentences or tokens, we propose the former solution since it represent a good compromise between tokens

and the full text), while concepts are arranged along the vertical axis, according to their temporal order. Moreover, as the chart incorporates data taken from three different sources (the output of the burst algorithm, the gold dataset and the textbook itself), we can use it to perform further kinds of analysis and textbook exploration. A textbook can be in fact viewed as a sequence of elements (concept, topics). [109] proposed to build intelligent Information Retrieval interfaces bases on a similar idea, so that an iconic representation could display a full document as a rectangle (tilebar) containing patterns of term distributions according to several spatial-temporal configurations (e.g., disjoint, local co-occurrence, global discussion of one or both terms).

- **Clusters Visualisation**. This visualisation helps to differentiate clusters of concepts as they have been recognised by a community detection algorithm executed in the graph. Intuitively these clusters shows the membership of a concept within a thematic unit (e.g., concepts related to network security, or to network classification, and so on), allowing to identify interesting elements (for example, bridging concepts, i.e. concepts that connect two different topics or textbook section). Clustering can be visualised with different techniques (graph with coloured nodes, dendrograms).

CHAPTER 5

DEVELOPMENT AND EXPERIMENTS

In this chapter we describe the process and experiments that led to the definition of the PRET framework. Each sub-chapter is focused on a module of the framework, its development process and the related published papers. Since other PhD theses deal with some of the experiments carried out by using the framework modules, the present thesis reports only those where the author mostly worked at the design and development. The development of each module was not a linear process, even though we struggled to keep the development stages of the framework as consistent as possible with the logical order in which a final user would most probably use the different modules of the framework (i.e. he would upload a text, annotate it, merge it with other annotations, compute agreement, analyse it and visualise it, use it to train or evaluate algorithms).

5.1 Text annotation

5.1.1 Starting the prerequisite annotation protocol

The framework described in the previous chapter originated from an initial study conducted on PR manual annotation, whose results were firstly reported in [12]. On that occasion, with the aim of supporting the process of manual annotation and analysis of PR relations, we defined a domain independent methodology for annotation. We then analyzed the resource resulted from the annotation process and addressed the problems that emerged in the annotation task and in the resource itself. This brought to the

design and development of a first prototype of the annotation module to support and validate human annotations. The resource was called PRET, which later gave the name to the whole framework. Note however that in the following sections we will refer to this resource as DATASET-1. The resource was a dataset annotated with prerequisite relations between educational concepts extracted from a Computer Science textbook. More precisely, 4 experts annotated the text with PR relations and their individual annotations were then combined. We came to build a resource, define a methodology and later develop a tool for manual prerequisite annotation since we observed some critical issues emerging during the annotation task, confirmed also by the literature. In particular, we noticed that experts encountered several difficulties to identify PR relations when they had to address the task starting from a list containing all possible pairs of concepts in a pre-arranged manner. Moreover, this design of the task was reported as very limiting because of the impossibility to add new concepts. We also noted that annotators often used their background knowledge about the domain instead of capturing what was written in the textbook. In order to face these issues, we designed a standalone tool to support and facilitate experts during the phases of manual annotation, textual pre-processing and analysis of its output.

The resource presented in [12] was constructed in two main steps: first we exploited computational linguistics methods to extract relevant terms from a textbook[1], then we asked humans to manually identify and annotate prerequisite relations between educational concepts. The annotation task consisted of making explicit the prerequisite relations between two distinct concepts if the relation can be somehow inferred from the text. As already said, we describe a concept as a domain–specific term denoting domain entities expressed by either single nominal terms (e.g. *internet, network, software*) or complex nominal structures with modifiers (e.g. *malicious software, trojan horse, HyperText Document*). Figure 2.1, in section 2.2.1, shows a sample of the concept map resulting from this annotation. For instance, according to this dataset, an example of prerequisite relation is *network* is a prerequisite of *internet*, since a student has to know *network* before learning *internet*.

In the following, we report the details regarding this initial study: the approach we used for the identification of concepts, the selection of annotators and the annotation task. Finally we present the characteristics of the dataset we obtained at the end of the process. In the next section (5.1.2) we will instead discuss the agreement computed on

[1]For the annotation we used chapter 4 from the Computer Science textbook "*Computer Science: An Overview*" [37].

this dataset for each pair of annotators, together with other statistics about the data.

Concept identification. Our methodology for prerequisite annotation requires concepts to be extracted from educational materials, that we broadly define Document (D), and provided to annotators. Although we are conscious that a concept, as mental structure, might entail multiple terms, we simplify the problem of concept identification assuming that each relevant term of D represents a concept [165]. Thus, our list of concepts is a terminology T of domain–specific terms (either single or complex nominal structures), where each concept corresponds to a single term in T, and terms are ordered according to their first appearance in D.

For the task of prerequisite annotation, it is irrelevant whether concepts are manually annotated, automatically or semi–automatically extracted. In our case, to build the resource, we extracted concepts with an automatic procedure. To identify our terminology T, we relied on Text-To-Knowledge (T2K^2) [69], a software platform developed at the Institute of Computational Linguistics A. Zampolli of the CNR in Pisa. T2K^2 exploits Natural Language Processing, statistical text analysis and machine learning. T2K^2 encompasses two main sets of modules, respectively devoted to carry out the linguistic pre-processing of the acquired corpus and to extract and organize domain knowledge from linguistically enriched (i.e. annotated) texts [3]. Each section of the considered textbook was automatically enriched with linguistic information at increasingly complex levels of analysis (sentence splitting, tokenization, Part-Of-Speech tagging and lemmatization). According to the methodology described in [32], the automatically POS-tagged and lemmatized input text is searched for candidate domain-specific terms denoting domain entities expressed by either single nominal terms (e.g. *internet*, *network*, *software*) or complex nominal structures with modifiers (typically, adjectival and prepositional modifiers). The latter are retrieved on the basis of a set of POS patterns (e.g. adjective+noun, noun+preposition+noun) encoding morpho-syntactic templates for multi-word terms (e.g. *Internet Protocol*, *eXtensible Markup Language*, *client/server model*) [3]. The domain relevance of both single and multi-word terms included in the extracted list T is weighted on the basis of the C-NC Value [86], that measures how much a term is likely to be conceptually independent from the context in which it appears. Accordingly, a higher ranking is assigned to those terms that are more relevant for the domain of D.

We applied T2K^2 to a text of 20,378 tokens distributed over 751 sentences. 185 terms were recognized as concepts of the domain (around 20% of the total number of nouns in the corpus). As expected, the extracted terminology contained both single nominal structures, such as *computer, network* and *software*, and complex nominal structures

with modifiers, like *hypertext transfer protocol, world wide web* and *hypertext markup language*. For this PR annotation experiment, the set of terms did not go through any post–processing phase.

Starting the PR annotation protocol. The analytical process of annotating PR demands to deal with ambiguity, and annotators are required to make substantial efforts towards common sense making during an interpretative and heuristic act. For this reason, starting with the preliminary phases for defining the annotation task, we used coding procedures that are commonly found in qualitative researches and fields such as social sciences, where categories often fall within fuzzy boundaries [191, 216]. We claim that such a coding approach can be also fruitfully applied to PR annotation, since this relation is often hard to define at best and its understanding is not perfect yet. Our motivation comes from the observation that in many cases there is a direct relationship between the development of a coding system and the evolution of a shared understanding of the phenomenon [238]. Coding procedures have been recently applied in textbook concept annotation (i.e. not relations) for knowledge engineering purposes [233]. In our case, the use of similar strategies for PR annotation eventually led us to a more shared vision of how the task of PR annotation could be treated, therefore an annotation protocol that specifies guidelines, annotators training, pre-annotation data gathering and post-annotation analysis. Annotation task (described in the present chapter), as well as annotation guidelines and eliciting questions (see appendix B), are the results of coding procedures that went through multiple stages, requiring iterative cycles of exploratory annotations and collaborative efforts towards a consensual treatment of our research issue. Even before the annotation of the initial study we collected and coded essential information about the corpus to annotate (e.g. level of knowledge required to the reader in order to understand, reasons why this textbook has been proposed and consensually accepted) and participants (i.e. basic demographic characteristics, their level of expertise about the domain and familiarity with annotation tasks in general). During the initial coding, criteria have been discussed within the research group to decide how to divide textual data in discrete parts, i.e. the level of granularity at which PR should be coded in the text (e.g. full-text-level, paragraph-by-paragraph or sentence-by-sentence coding). Based on mutual agreement, a concept has been defined as a keyphrase, made up of one or more terms, representing a key component in the domain of the textbook. Starting with this first phase a shared coding document has been used to keep a record of all emergent ideas and categories, along with their descriptions, inclusion and exclusion criteria, typical and atypical examples, discussions about different proposed methods of

annotation.

Annotators selection. The role of annotators is fundamental for obtaining a gold dataset that represents the pedagogical relations expressed in educational materials. Consequently, the choice of annotators is crucial. As mentioned above, in the literature annotators are often domain experts [78, 135, 136] or students with some knowledge in that domain [169, 234]. Based on our experience with different types of annotators, we suggest that they should have enough domain knowledge to understand the content of the educational material. Otherwise, the annotation can be distorted by a lack of comprehension of the relations between concepts. On the other hand, experts should not rely on their background knowledge to identify relations, since the goal of the annotation is to capture the knowledge embodied in the educational resource. To build the dataset presented here we recruited 6 annotators among professors and PhD students working in fields related to Computer Science (note however that 2 of them were eventually revealed not to have enough knowledge for the task).

Annotation task. To keep the annotation as uniform as possible, we provided the annotators with suggestions on how to perform the task, together with the book chapter and the terminology extracted from it. Considering the supplied material, we asked annotators to trust the text, considering only pairs of distinct concepts in T and annotating the existence of a prerequisite relation between the two concepts only if derivable from D. With this method, annotators should read the text and, for each new concept (i.e. never mentioned in the previous lines), identify all its prerequisites. On the other hand, if no prerequisite can be identified, they should not enter any annotation. We also wanted to preserve the properties of the prerequisite relation (as defined in 2.2.1), so we asked annotators to respect them. In particular, we specified not to annotate self–prerequisites, since this would obviously violate the irreflexive property. Considering the topology of a concept map (let us remember that this is a DAG), we also asked annotators not to enter cycles in the annotation. To better understand this point, consider the concept map in Figure 2.1: having inserted a prerequisite relation between *computer* and *network* and between *network* and *internet*, entering a relation where *internet* is prerequisite of *computer* would create a cycle. For the sake of uniformity, we also ask to avoid adding transitive relations, i.e. do not enter a relation between two concepts A and B if the textbook actually explain them using an intermediate concept C (thus, annotate only $A \prec C$ and $C \prec B$, not $A \prec B$). The output of the annotation of each annotator is an *enriched terminology*: a set of concepts paired and enhanced with the prerequisite relation. The enriched terminology can be used to create a concept map where each concept is a node

while edges are prerequisite relations identified by humans (see Figure 2.1).

Annotation instrument. As a support for the annotation of this dataset, the experts used a $|T| \times |T|$ matrix M built from the terminology T, where they entered a binary value in the cell $M_{A,B}$ to indicate the presence of a prerequisite relation $A \prec B$.

The dataset. The gold dataset G consists of 34,225 concept pairs obtained by all possible combinations of the elements in the concepts set excluding self–prerequisites (i.e. $|G| = |T| \times |T| - |T|$). Pairs vary with respect to the relation weight w, computed for each pair (A,B) by dividing the number of annotators that inserted $A \prec B$ by the total number of annotators. Only for 1.54% (= 526) of the pairs in G we could observe $w > 0$ (i.e. the prerequisite relation was annotated by at least one annotator). Details about the distribution of prerequisite relations and respective weights are reported in Table 5.1. Interestingly, 55.70% (= 293) of the concept pairs with $w > 0$ were identified by only one annotator. This number shows how hard is for humans to agree on what and where a prerequisite is.

Evolution of the PR annotation protocol. As can be seen in the previous paragraph, the first cycle of annotation revealed the issue of disagreement, which is actually not unusual in tasks where two or more codes (in our case PR and not PR) could be equally and validly applied to the same passage of text. In such cases it is therefore difficult to decide *a priori* which label is correct (see [191] for similar situations where simultaneous coding applies). While this situation is a direct consequence of the intrinsically ambiguous nature of PR relation, it also leaves researchers difficulties when their purpose is to build a robust dataset for algorithm training. In other words, according to our purpose (analysis/understanding of the phenomenon or algorithm training) different outcomes may be more desirable, i.e. a qualitatively rich annotation or an unanimous agreement. We observe in particular that inter-agreement metrics such as k are well designed for categorical data and quantitative analyses, but do not comfortably fit in annotations involving more qualitative aspects, such as PR annotation. More in general, in qualitative research there is not always a standard minimal value of intercoder agreement that has to be reached among coders: high values can be taken as minimal thresholds by those methodologists that look for a statistic evidence, while some others may even question the utility and feasibility of agreement when data and tasks involve interpretive and subjective processes [106, 191, 192]. After the emergence of this inter-agreement issue in PR annotation, we kept on using inter-annotator agreement metrics as a benchmark to monitor the process and to study how the agreement changes according to different settings. At the same time, in line with [24, 192], we also explored social negotiation

Relation Type	**Weight**	**Count (%)**
Non–prerequisite	0	33,699 (98.46%)
Prerequisite	> 0	526 (1.54%)
Total number of pairs		34,225

Annotators	**Weight**	**Count (%)**
1 annotator	0.25	293 (55.70%)
2 annotators	0.50	131 (24.90%)
3 annotators	0.75	75 (14.26%)
4 annotators	1	27 (5.13%)
Total number of pairs		526

Table 5.1: Relations and weight distribution in the dataset.

practices for increasing group consensus, with the intention of preserving a good trade-off between validity (i.e. the degree to which relevant features according to annotators are well expressed in the coded data, see [103]) and inter-coder agreement (whose achievement may in some cases bring to simplifications and eventually compromise validity, see [192]). Think aloud sessions helped to negotiate disagreement between annotators, that were asked to compare their different points of views, provide explications of their judgements and verbalise the reasoning behind their choices. Including and excluding criteria emerged during these sessions are reported in the code book in appendix A (i.e. code book). Note that the text that was annotated during such iterations focused on computer networks, but in the code book we transferred the examples emerged from discussions among experts during post-annotation sessions to different topics within the same domain (i.e. other subtopics of computer science). This was motivated by the need to avoid suggesting particular relations to other annotators during the next iterations and, at the same time, provide them with familiar examples, i.e. belonging to their domain of expertise. More specifically, examples listed in the code book are drawn from textbook chapters that deal with computer architectures, programming languages, data representation, software engineering, computer graphics, algorithms.

5.1.2 Towards the creation of datasets within PRET Framework

5.1.2.1 Development of the Annotation Module

The study described in the previous section revealed some critical issues affecting the manual annotation of prerequisites in actual texts:

- **Agreement.** Quantitative analysis conducted on data annotated by different experts revealed the issue of low agreement among annotators, as can be guessed from the aforementioned data (i.e. 55.70% of relations had the minimum possible agreement since were identified by only one annotator).

- **Annotation validation**. Analysing the annotated data, we noticed that human experts are not immune from making mistakes and violating the supplied recommendations regarding the prerequisite properties.

Based on this experience and the encountered issues, we developed and provided a language and domain independent tool which aims on the one hand to support and validate the annotation process and on the other hand it is thought to be an annotation module integrated in a larger framework which allows also to perform annotation analysis, agreement computation, extraction and visualisation. All the main features of the tool (a screenshot is shown in Fig. 5.1) have been conceived taking into account real problems encountered while building the dataset described above and confirmed by the analysis of the related literature (see the issues discussed in section 3.1). Thus, this tool is highly valuable for annotators because specifically addresses annotators' needs. To support the annotation, the user is provided with the terminology T as a list of concepts ordered according to their first occurrence in the text. This is done in order to give the annotator an overview of the context in which the concept occurs. We observed that the textual context plays a crucial role in deciding which concepts are prerequisites of the one under observation, so for each term we show the list of other terms with visual indication of the progress in the text. Additionally, the tool validates the concept map derived from the annotation introducing controls that prevent the annotator from making errors (e.g. cycles, reflexive relations, symmetric relations). Lastly, the justified suspicion of a low agreement pushed us to use appropriate metrics to quantify such value, investigate also the possible reasons. We further investigate this aspect in section 5.3.

5.1.2.2 Creation of new datasets using the Annotation Module

As we asked annotators to take part to the next iterations of the evolving annotation protocol, we produced new datasets which differ in characteristics and creation criteria. In the present section we present two annotated datasets created after DATASET-1, i.e. DATASET-2 and DATASET-3.

DATASET-2. In this new dataset, the five experts of DATASET-1 were asked to re–annotate the same text indicating any prerequisite concept of each relevant term

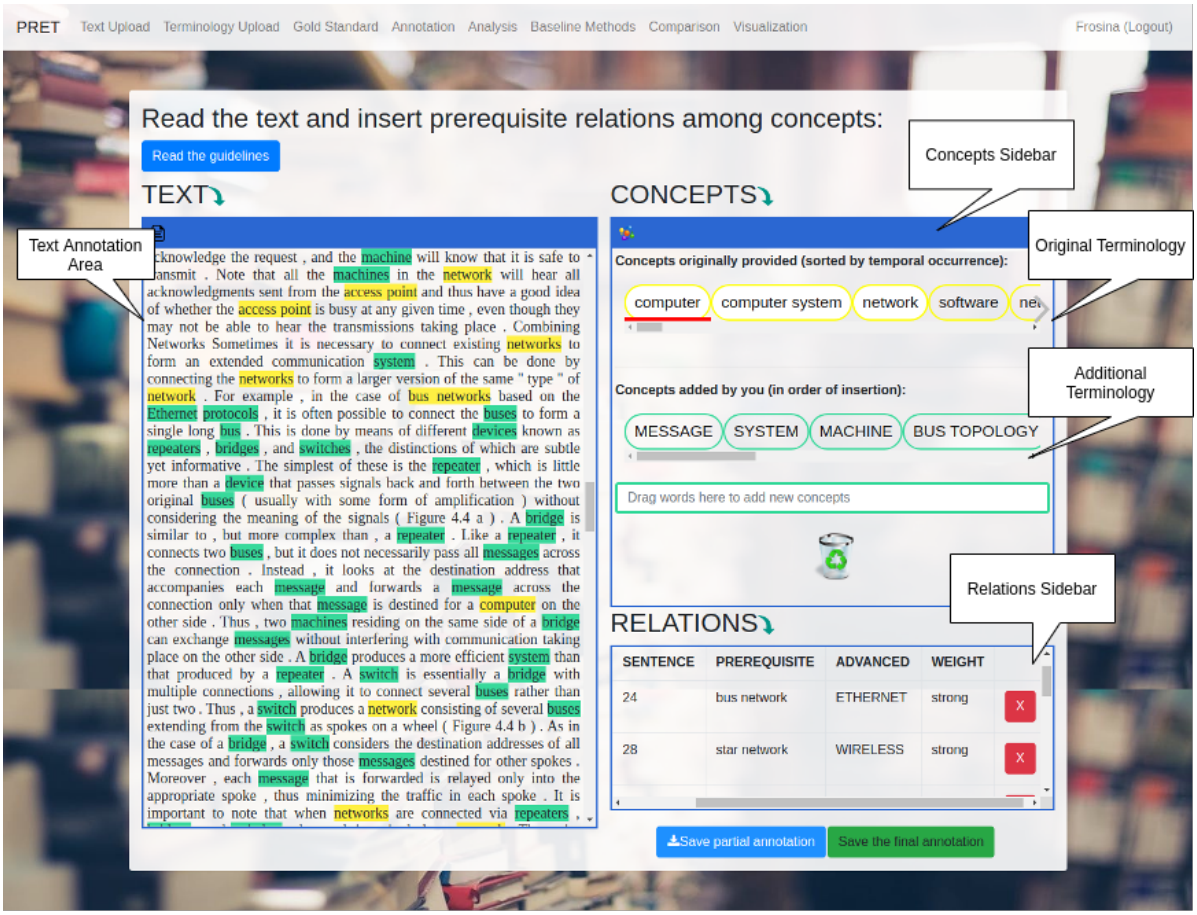

Figure 5.1: Annotation module interface

appearing in the text. Explicit guidelines were provided, as well as eliciting questions for helping them to understand what a prerequisite relation is (both guidelines and questions can be found in the appendix, see B). In addition, during this iteration of the annotation protocol we introduced a simple coding schema for marking the magnitude of the relation (strong or weak) according to the annotators. The set of relevant terms was extracted with the same automatic strategy described in [12], but this time the list was manually validated by three experts in order to identify a commonly agreed-on set of concepts, which resulted in a terminology of 132 concepts. Besides these terms, each expert could independently add new concepts to the terminology when annotating the text if he/she regards them as relevant. In fact, as said in chapter 4, the annotation module allows the creation of the terminology during the annotation process, thus enabling also the enlargement of the terminology. Consequently, experts produced different sets of concept pairs annotated with prerequisite relations since 221 new concepts were manually added

during the annotation process (for a total of 353 concepts in the dataset). The inserted relations obtained different agreement values (i.e. number of raters that identified the relation). More precisely, the manual annotation resulted in 25 pairs annotated by all five experts, 46 annotated by four experts, 83 by three, 214 by two and 698 by only one annotator, for a total of 1,066 pairs. As in the case of the first dataset, it is highly possible that the high divergence in experts' opinion for pair creation reflects a difficulty in determining what defines a prerequisite relation between concepts.

DATASET-3. For this dataset we recruited four new annotators among students enrolled in a Computer Engineering master's degree program. All of them were familiar with the topic of the textbook and they had a comparable background knowledge in the domain. Before starting the annotation task we carried out a training phase to introduce the task of PR annotation, showing them the guidelines and addressing major doubts about the annotation protocol. This training phase was essential to address unclear aspects of the annotation before creating the final version. Unlike DATASET-2, in the present annotation we asked experts to create PR pairs only among those concepts that were originally provided them and obtained by means of a revised (i.e. semi-automatic) extraction. The reasons behind this choice can be understood if we consider the different nature of study that one may want to conduct on an annotated dataset. By allowing experts to independently add new concepts to the terminology we generally obtain a sparser dataset with a lower agreement between annotators, due to the fact that each annotator had the possibility to insert PR relations involving a particular concept that was potentially present only in his terminology. This aspect does not necessarily represent a problem if our goal is to study a phenomenon and perform corpora analysis. In such cases, in fact, letting annotators express their intuitions as freely as possible can be a strategy to obtain rich annotated resources, which can be then analysed in depth or collaboratively discussed by annotators to understand the problem that they are trying to capture. The same approach could be instead discouraged if the goal is to obtain a more compact dataset for training or evaluating automatic systems, since the resulting dataset will be probably affected by sparsity due to the high amount of subjectivity introduced during the annotation. In such cases a more suitable scenario may be represented by using a shared and reliable terminology and ask annotators to focus on that.

5.2 Annotation Analysis

The set of functionalities provided in the analysis module allows to inspect datasets from different perspectives (quantitatively, linguistically and using graph metrics). We will describe in this section some of the analysis we made on the datasets.

Quantitative Analysis. A Data Synthesis provides, given a dataset, the number of concepts, number of relations, number and list of non–prerequisite relations, transitive relations, and conflicting relations between annotators. This is the first set of functionalities, that have been built. They were used for instance for obtaining the quantitative data reported above DATASET-1.

Table 5.2 shows the results obtained by running quantitative analysis on the revised DATASET-3. All annotators worked on the same text, thus the size of the annotated corpus is the same for all of them. On the other hand their annotations vary in terms of number of added relations. For example, annotator 2 and 3 are the only two that added more than 400 unique PRs (i.e. each concept pair counted only once), while the other three annotators inserted a comparable amount and relations. Also the distribution of relations' weight splits the annotators in two groups: annotator 1 and 3 on one side, with around 75% of PRs indicated as *strong*, and annotators 2, 4 and 5 on the other side, with *strong* PRs constituting more than 90%. For what concerns transitive relations, all annotators adopted similar criteria for adding them since in all annotations these relations correspond to around 25% of total PRs.

These results identify some clear similarities between the annotations, but more sophisticated measures are needed to better describe the characteristics of each annotation.

	Annot1	**Annot2**	**Annot3**	**Annot4**
# of Sentences	751	751	751	751
# of Tokens	20,378	20,378	20,378	20,378
# of PRs	130	237	185	144
# of Terms	140	140	140	140
% of Strong PRs	93.08%	73.00%	83.24%	82.64%
% of Weak PRs	6.92%	27.00%	16.76%	17.36%
% of Transitive PRs	3.88%	24.68%	18.13%	16.78%

Table 5.2: Summary of the results obtained using quantitative analysis on the revised DATASET-3 for each annotator.

Graph Analysis. We made use of graph analysis algorithms since the first iteration of the annotation protocol, considering peculiar features of PRs graphs such as relation direction, roots and leafs comparison and transitivity. Besides computing basic metrics in the graph, our main goal was to identify annotators mistakes. The analysis carried out on DATASET-1, i.e. before applying validation checks against possible mistakes, highlighted some critical issues. Transitive relations, for instance, were explicitly annotated, and some cycles were erroneously added in the dataset, violating the instructions. While cycles were due to distraction, transitive relations represent a harder case to deal with. Annotators do not easily recognised them per se, especially when a broader or generic term is involved (e.g. *computer, software, machine*). In other words, even if annotators are explicitly instructed not to add any transitive relation, they can be nonetheless prone to insert relations such as *computer* $\prec$ *local area network* when they have already inserted *computer* $\prec$ *network* and *network* $\prec$ *local area network*. Given the complexity of an annotation task based on the text flow, it is hard for annotators to be consistent with the same behaviour throughout all the annotation (i.e. do not annotate any or annotate all transitives). As a result, each annotator introduces a certain number or different transitives. In order to study how these issues impact the dataset, each annotation was checked for searching cycles and transitive relations, obtaining different dataset variations (in addition to the original annotation). This procedure was conducted using graph analysis on the concept map derived from the enriched terminology of each annotator. More in specific, we operated on cycles and transitive relations: in some variations, the latter were added if the pair of concepts in the concept map is connected by a path shorter than a certain threshold (defined by considering the concept map diameter), while cycles were either preserved or removed depending on the variation we wanted to obtain. We eventually obtained the following annotation variations:

- *no cycles* (removing cycles);
- *cycles and transitive* (preserving cycles and adding transitive relations);
- *cycles and non–transitive* (preserving cycles and keeping only direct links);
- *no cycles and transitive* (removing cycles and adding transitivity);
- *no cycles and non–transitive* (removing both cycles and transitivity).

We used graph analysis also in DATASET-2 to obtain a high-level description of the annotations. For this dataset we analysed on relation direction and root/leaf nodes. The

analysis performed specifically on the direction of the prerequisite relation revealed a high agreement between the annotators: only 44 relation of all annotated relations from all the annotators reveal a direction disagreement. A vertex property study based on in-degree and out-degree of the annotators graphs supported that: the concepts with high in-degree (e.g., *link layer*, *bridge*, *proxy server*, *firewall*, *uniform resource locator*) are advanced concepts which require much background knowledge in order to be learned well. On the other hand, concepts with high out-degree are more fundamental concepts (e.g., *computer*, *network*, *software*, *server, protocol*). Finally, concepts with both high in-degree and out-degree such as as *internet* and *router* can be interpreted as concepts that mark/represent the domain.

Context Analysis. When analysing the textbook in search for prerequisite relations, we notice that those relations assume different forms and linguistic realisations. The framework, in particular the interface Prerequisite-In-Context (PIC, inside the analysis module, see Fig. 5.2), allows the user to query an enriched dataset and look for particular instances of the phenomenon. PIC gives information on the linguistic features (e.g. text, lemmas, POS tags and parsing) associated with the context where the relation has been inserted by the annotators (i.e. the sentence where the relation occurs, as well as the previous and next sentence, see Fig. 5.3).

With the aim of supporting researchers in their effort to understand PR, we propose to examine the text and look for a set of categories that may reveal significant or recurrent phenomena. The categories that we propose here emerged after the second iteration of the annotation protocol (see section 5.1.2.2), when we expanded the coding procedure by including a post-annotation structural subcoding of the annotated text. More in detail, we asked annotators to re-read segments of text that were coded with PR and classify them in order to describe which kind of PR could be observed in that segment of text according to their opinions. When comparing their labels, similar classes were then merged, while the less significant were abandoned. As section 5.3 will describe, a similar subcoding procedure was also performed when annotators revised their own annotation, using inter-annotator agreement to identify candidate wrong relations and then categorising annotation errors into a system of second-order classes. Once a consensus was reached regarding the schema to follow during annotation analysis, annotations have been analysed using the resulting taxonomy. This is based on categories that can be divided into pedagogical and linguistic categories. The categories proposed for pedagogical relations tend to widely borrow from the instructional frameworks we reviewed in 2.1.2

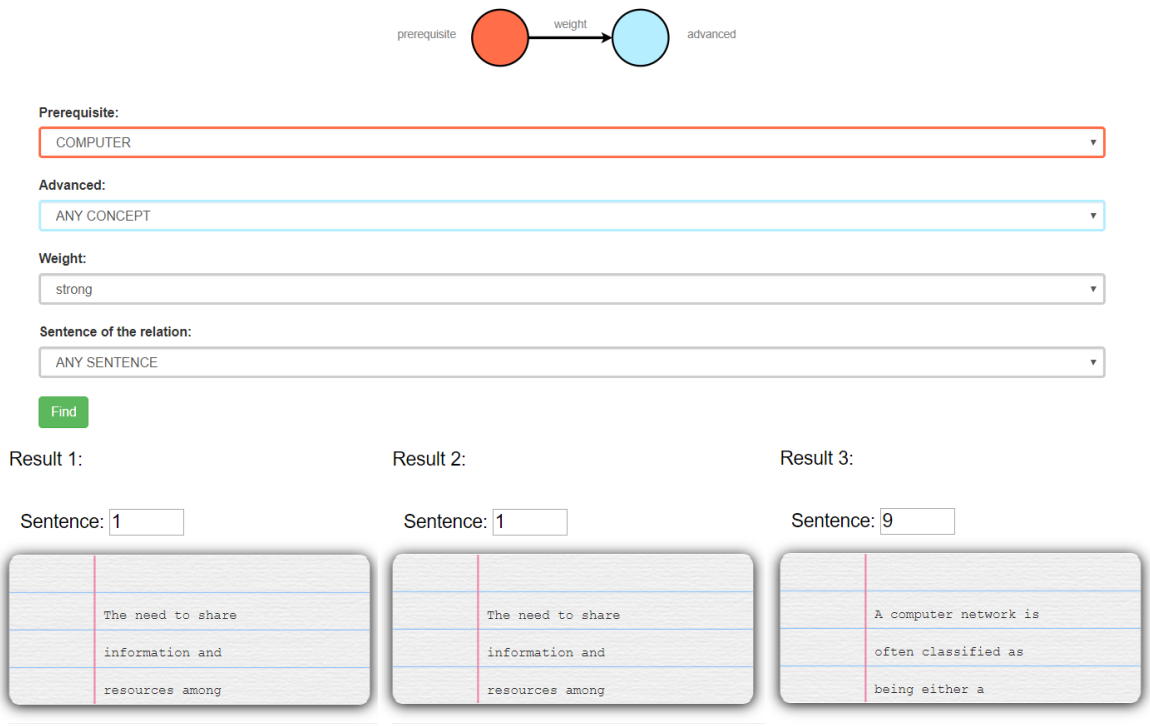

Figure 5.2: Prerequisite In Context query interface

and also encompass some notorious semantic relations. Namely, following such schema, we analysed whether the two concepts related by a PR in the text belong to a:

- hyponym-hypernym relation
- holonym-meronym relation
- procedural relation
- causal relation
- list of conceptual items

Concerning the linguistic features, we can define the selected categories as follows. Given a relation $X \prec Y$ annotated in the sentence $S[i]$ (where S is the list of all sentences constituting the full text), we investigate whether:

- the first occurrence of Y takes place in $S[i \pm 1]$;

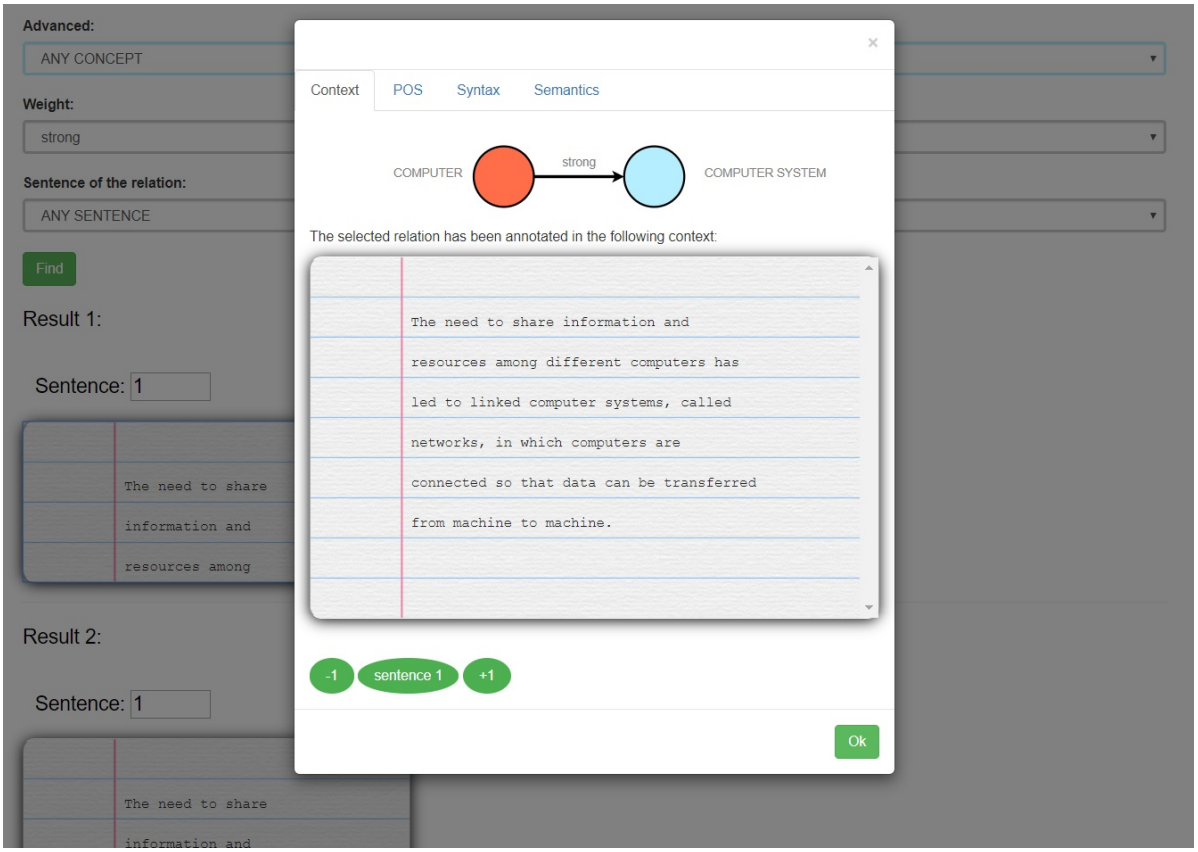

Figure 5.3: Prerequisite In Context results analysis

- the definition of Y is given in the context of PR annotation (i.e. one sentence in the range $S[i \pm 1]$ represents a definitional sentence for the advanced concept);
- some lexico-syntactic pattern is expressed in $S[i \pm 1]$;
- X is the lexical head or a modifier of a complex nominal structure (e.g. *software* $\prec$ *malicious software*, or *internet* $\prec$ *internet infrastructure*).

An analysis conducted on the most frequently recognised relations (around 20% of PRs) shows that almost half of the observed relations (49,60%) corresponds to a lexical taxonomic relation, such as , hypernymy/hyponymy and holonymy/meronymy. This can happen when the text deals with a topic in detail and/or when one of the two concepts is a multi-word term and the other is its lexical head. The remaining PRs mostly fall into the very interesting (but critical to analyse) grey area of flow relations. In other words, when an explanation is taking place, concepts are still presented in the text according to a sequence: in such cases, even without an expressed linguistic phenomenon

revealing a PR, experts do recognise the concepts as being connected by a prerequisite relation, which however is not possible to categorise into a particular pattern or schema. Interestingly, only 60% of those pairs reflect the order of appearance of the concepts in the text (i.e. only in 60% of cases, previously mentioned concepts become prerequisite of the subsequent ones).

An interesting phenomena is observable when two concepts establishes prerequisite relations because they form a list of co-related concepts. As recollected in 2.1.2, lists represent a possible way to organise in a sequential way two or more concepts that fall into the same category but are not related in an ontologically hierarchical way. A closer look to one example derived from the annotated corpus will clarify this phenomenon (the excerpt discusses the family of devices used to connect different networks, and the first occurrences of the relevant concepts are underlined by us):

> *The simplest of these [devices] is the repeater, which is little more than a device that passes signals back and forth between the two original buses (usually with some form of amplification) without considering the meaning of the signals. A bridge is similar to, but more complex than, a repeater. Like a repeater, it connects two buses, but it does not necessarily pass all messages across the connection. [...] The connection between networks to form an internet is handled by devices known as routers, which are special purpose computers used for forwarding messages. Note that the task of a router is different from that of repeaters, bridges, and switches in that routers provide links between networks while allowing each network to maintain its unique internal characteristics.* [37]

In the quoted excerpt, *repeater*, *bridge* and *router* denote equally a sort of device with a similar function. In an ontological representation these three concepts would be hence arguably encoded as sibling nodes, since they all belong to the same level, possibly children of *device*. This last concept, after all, is lexically a hypernym of all three concepts, and in the text plays the role of a primary notion, i.e. a concept that is never really explained by the author because its presence in the student's prior knowledge is taken for granted. Despite the ontological non-hierarchical organisation, during the flow of the exposition, the author naturally presents these concepts in a sequential order, offering for each of them a definition based on similarities and differences with respect to the others. As a result, understanding one of them, for the reader, becomes very useful to understand the next in line. This arguably explains why annotators inserted similar

prerequisite relations between such co-related concepts. It also gives us a further hint about how PR are harder to model in textbooks by relying only on external sources of knowledge that are independent from the textbook.

Contrary to what may be derived from theoretical classifications reviewed in section 2.1.2, the analysed sample did not show any case of causal relations (e.g. concept A is caused by, or is an effect of, concept B). This could be however justified by the characteristics of corpus domain, while in other fields there could be a larger coverage of causal links (e.g. physics, history, medicine, among others). The same holds for procedural relations (e.g. A represents a step in a procedure that is required before B). This kind of relations are rare in our sample, but they could play a much more important role in domains where procedures are more frequent, such as for instance statistics, or even some different topics of the same domain of Computer Science (e.g., algorithms). Further, cross-domain, investigations are necessary to confirm similar intuitions though.

5.3 Combination of annotations

A gold standard is created by combining different annotations produced by individual annotators and regarding the same text. Individual annotations are thus ready to be combined in order to obtain a gold standard dataset. Before proceeding with the combination though, we evaluate annotations' homogeneity in pre- and post-revised individual annotations using inter-rater agreement metrics.

Annotators agreement evaluation. Our experience and the literature [78] show that human judgments about prerequisite identification can vary considerably, even when guidelines are provided. This can depend on several factors, including the intrinsic complexity of the prerequisite relation, the subjectivity of annotators and the type, complexity and expository style of the learning material. Although the agreement distribution (shown in Table 5.1) captures quite well the dataset characteristics and identifies potentially problematic areas in the annotation, more sophisticated coefficients of agreement are available in the literature: the three best-known are S [25], π [196] and κ [62], in addition to generalisations based on their formulas. Related work addressed the task of prerequisite annotation between pairs of concepts as a labelling annotation task using a small set of predefined categories: given a pair of concepts (*item*) decide whether they show a prerequisite relation [54, 171, 236]. This annotation strategy is highly similar to other popular annotation tasks in NLP and corpus linguistics (e.g. sentiment and polarity annotation, error annotation).

In our case, annotators have to build the concept pairs by themselves and only when a prerequisite relation is identified, while all negative relation are left unexpressed (they still can be automatically derived by generating all possible concept pairs that were not annotated by any expert and labelling them as “non prerequisite”). This fact has strong implications not only when designing annotation guidelines, but also on how we compute agreement between raters since the number of negative relations is one of the parameters considered by all coefficients. In the following paragraphs we will describe how we evaluated annotators’ agreement within PRET in DATASET-1, DATASET-2 and DATASET-3, which derived from the application of the revision protocol.

1. Computing Agreement in DATASET-1. We first computed agreement on the unvalidated annotated dataset (i.e. preserving all the erroneously inserted cycles as well as without modifying transitives relations). Following [15], we computed the agreement between multiple annotators using Fleiss’ k [84] and between pairs of annotators using Cohen’s k [62]. Using the scale defined by [126], Fleiss’ k values show *fair agreement*, suggesting that prerequisite annotation is difficult. Similar tasks obtained comparable or lower values, confirming our hypothesis: [99] measured the agreement as Pearson Correlation obtaining 36%, while [78] and [54] obtained respectively 30% and 19% of Fleiss’ k. Next we investigated how agreement varies when cycles and transitives are handled, thus we computed these metrics on the different versions of the dataset that we created by traversing the graph (listed in 5.2, paragraph "Graph Analysis"). Compared to the other variations, removing cycles and adding transitive relations showed the highest improvement on the agreement, also for pairs of annotators (Table 5.3). Our results suggest that different levels of domain knowledge possessed by the experts entails different annotations and values of agreement, confirming previous results [99]: lower agreement can be observed when annotator 4 (quasi–expert) is involved, possibly due to the lower competence level if compared to the other annotators. Annotator 4 is also the one who considered the highest number of transitive relations, i.e. this annotator considered a higher number of prerequisites for each concept, producing a more connected concept map. On the other hand, annotators with more experience show even *moderate* (pairs A1-A3 and A2-A3) or *substantial agreement* (pair A2-A3 for the variation). Adding transitive relations and removing cycles generally improves the agreement values also when we consider pairs: we notice an increase of 8.35 points for A1-A2. The only exception is observed for the pair A1-A3, which experienced a decrease of almost 7 points. The cause is though to be the number of transitive relations considered by annotator 3, which

Metric		Orig.	No Cycl. & Trans.	Diff
Fleiss's k	All raters	38.50%	**39.94%**	+1.44
Cohen's k	A1-A2	34.46%	42.81%	**+8.35**
	A1-A3	57.80%	50.84%	**-6.96**
	A1-A4	37.59%	39.29%	+1.70
	A2-A3	56.50%	63.62%	+7.12
	A2-A4	28.02%	29.42%	+1.40
	A3-A4	25.35%	25.71%	+0.36

Table 5.3: Agreement values and differences for two annotation variations for DATASET-1.

is around one third of the transitive relations annotated by annotator 1: the validation creates more distance between the two annotations reducing the agreement.

2. Computing Agreement in DATASET-2. We computed two agreement values, both in terms of κ score [62], between all pairs of annotators. In one case, we took into account all 353 concepts included in the final terminology: we considered only explicitly created concept pairs, and for each annotator we counted as positive pairs manually inserted relations and as negative pairs those pairs annotated by one of the other raters but not by himself. In this case, the average κ score between all pairs of raters is 0.11. In the other setting, we took into account only the 132 automatically extracted concepts since those are shared among all raters. In this case, we generated all possible pairs of concepts and assigned them to each annotator, considering them as positive if the rater assigned a prerequisite relation to the pair, and negative otherwise. Again, we computed agreement as the average κscore between all pairs of raters, obtaining *substantial agreement* (0.43).

For the purposes of this use case, we did not take into account relation weights when creating the gold dataset. Nevertheless, we noticed that the distribution of relation weights is significantly unbalanced: three raters out of five assigned a strong weight to more than 90% of annotated relations, while the remaining two raters assigned a strong weight to around 75% of the relations. This might indicate either that prerequisite relations are *strong* by definition or that the guidelines need to make a clearer distinction of the value of these two labels. To address the latter intuition, the tool needs to be tested in different contexts and domains.

As said in section 4.2.3, the combination module includes functionalities for supporting (i) computation of agreement between annotations made by different experts, and

also (ii) dataset revision. The coding system for revision that we present here consists in (i) a protocol for conducting revision; (ii) a labelling system (i.e. a tagset) for identify annotations error types; (iii) a description of the categories of errors. As we explain below, this revision approach was an outcome of the iteration of the annotation protocol that led to the creation of DATASET-2. The same system was then used to revise also DATASET-3.

After completing their annotation, annotators of DATASET-2 carried out a revision process aimed to reconsider certain PRs that we can identify as candidate errors, i.e. PRs annotated only by one expert. The underlying assumption is that there is higher probability to find annotation errors among those relations inserted only by one annotator, while the higher the number of annotators the lower the probability that we are facing a wrong relation. Focusing on relations inserted only by a single expert provides thus a balance between coverage and feasibility of the revision. Each annotator revised the pairs created only by him/herself and was able to confirm the pair, deleting it or modifying its weight (from strong to weak or vice versa) after reading again the segment of text where the relation was identified. Annotators were asked to individually revise their annotations and then share their choices and motivation for error coding criteria during collaborative sessions. They hence jointly defined a schema for detecting and revising wrong relations in a dataset. The resulting revision schema consists in the following categories, that annotators must choose after re-reading the context of annotation:

1. *Annotation error*: the PR was inserted by mistake (i.e. distraction);

2. *Background knowledge*: the relation is not explicitly explained in the text and it derives instead from the annotator's knowledge about the topic;

3. *Wrong direction*: target and prerequisite concepts were inverted in the pair;

4. *Co-requisites*: although related, there is not a dependency relation between the two concepts;

5. *Too far*: the relation is too weak because there are too many concepts between the prerequisite and the target concept, thus the former is not strictly essential to understand the latter.

6. *Not a concept*: at least one of the concepts in the pair should not have been added as a concept (this label has to be used only when annotators have the ability to add concepts in addition to relations);

	REVISION		ERROR TYPES					
	#Rev PR (% over tot PR)	#Del PR (% over Rev PR)	Not a Concept	Background Knowledge	Too Far	Annotation Error	Wrong Direction	Co-PR
A1	175 (42.68%)	27 (15.43%)	37.04%	29.63%	3.70%	11.11%	7.41%	11.11%
A2	72 (25.35%)	16 (22.22%)	62.50%	0.00%	6.25%	12.50%	6.25%	12.50%
A3	190 (43.68%)	86 (45.26%)	72.94%	2.35%	17.65%	5.88%	0.00%	1.18%
A4	118 (42.30%)	44 (37.29%)	70.45%	9.09%	2.27%	13.64%	4.55%	0.00%
A5	109 (39.64%)	29 (26.61%)	55.17%	31.03%	10.34%	3.45%	0.00%	0.00%

Figure 5.4: Revision summary for DATASET-2. 'Revised' columns report the number (absolute and relative) of Rev(ised) and Del(eted) PRs for each annotator. 'Error Type' columns report for each annotator the percentage of deleted pairs assigned to each label.

Figure 5.4 reports a comparison between annotators regarding their revision. *Not a concept* is the most recurrent problem for all annotators: as examples, common-usage terms like *channel* and *system* were considered as domain concepts during annotation but then excluded during revision. Among other error types, *Background Knowledge* and *Too far* suggest a certain annotation style, i.e. the tendency to add inferred relations (either from the expert prior knowledge or from other parts of the text). On the contrary, the other three types are all somehow due to distraction. In this respect, we observe that annotator 1, 3 and 5 mostly revised their style, while annotator 2 and 4 were mostly distracted but did not questioned their choices.

Agreement is reported in 5.4, computed as Cohen's k scores on both the pre-revision and post-revision DATASET-2. For this iteration of the annotation protocol experts were provided with an initial terminology that they could individually expand by adding new concepts. Given this fact, we computed k scores considering both sets of concepts, i.e. S1 (agreement over all possible pairs of concepts among the combined concept set consisting of 354 concepts) and S2 (agreement over all possible pairs belonging only to the initial 132 concepts).

The final gold dataset was then created combining all the annotations and considering as positive pairs (i.e. showing a prerequisite relation) all pairs of concepts annotated by at least one expert after the revision (see section 4.2.3 for combination criteria). The combination of all five annotations produced a gold standard dataset composed of 353 concepts (221 new concepts were manually added to the terminology) and 1,066 PR relations. Obviously, not all 353 concepts were actually used in all annotations. In particular, 21.53% appear in only one annotation and 25.78% appear in all the five annotations, 115 (32.58%) concepts were used in two annotations, while only 11.05% and 9.06% were used in 3 and 4 annotations respectively. Also the PR relations obtained

		A1		**A2**		**A3**		**A4**		**A5**	
		O	*R*	*O*	*R*	*O*	*R*	*O*	*R*	*O*	*R*
A1	*S1*	1	1	0,30	0,12	0,48	0,15	0,23	0,09	0,27	0,11
	S2	1	1	0,57	0,08	0,55	0,08	0,61	0,11	0,55	0,09
A2	*S1*	0,30	0,12	1	1	0,35	0,32	0,32	0,35	0,41	0,37
	S2	0,57	0,08	1	1	0,50	0,45	0,62	0,45	0,74	0,44
A3	*S1*	0,48	0,15	0,35	0,32	1	1	0,39	0,27	0,30	0,31
	S2	0,55	0,08	0,50	0,45	1	1	0,70	0,39	0,55	0,44
A4	*S1*	0,23	0,09	0,32	0,35	0,39	0,27	1	1	0,36	0,30
	S2	0,61	0,11	0,62	0,45	0,70	0,39	1	1	0,73	0,36
A5	*S1*	0,27	0,11	0,41	0,37	0,30	0,31	0,36	0,30	1	1
	S2	0,55	0,09	0,74	0,44	0,55	0,44	0,73	0,36	1	1

Table 5.4: K scores obtained on the prior (O[riginal]) and post (R[evised]) revision annotations of DATASET-2.

different agreement values (i.e. number of raters that identified the relation). From a quantitative perspective, we observe that 25 pairs were annotated by all the five experts, 46 annotated by four experts, 83 by three, 214 by two and 698 by only one annotator.

3. Computing Agreement in DATASET-3. We evaluated annotations' homogeneity in pre- and post-revision dataset-3 using inter-rater agreement metrics in terms of k score. Considering that the annotation has been performed here by four annotators, we report both Cohen's and Fleiss' implementation of k scores: Cohen's k is computed between all pairs of annotations, while Fleiss' k accounts by definition for multiple annotations on the same set of items. Table 5.5 reports the values of Cohen's k obtained between all pairs of original and revised annotations. We can notice that all pairs obtained similar agreement values and generally the revision generated more homogeneous annotations. According to the interpretation of k given in [126], in fact we observe an average *moderate agreement* (0.58) among the original annotations, which improves to *fair agreement* (0.60) considering the revised annotations. Overall, these results are in line with those observed for other prerequisite annotated datasets and reflect the high subjectivity of the task, which results in generally low k values [54, 78]. On the other hand, Fleiss' k was 0.446 in the original dataset, while in the revised is 0.470.

Table 5.6 reports the outcome of the revision. Even though each annotator created a different amount of PRs, they all revised a comparable number of pairs (between 25 and 33% of the total annotation). Considering the number of modified, deleted and confirmed PRs, experts seem generally more prone to validate their choices by confirming, or at

	A1		**A2**		**A3**		**A4**	
	O	*R*	*O*	*R*	*O*	*R*	O	R
A1	1	1	**0,399**	0,466	0,450	0,461	0,431	0,468
A2	**0,399**	0,466	1	1	0,419	0,449	0,412	0,454
A3	0,450	0,461	0,419	0,449	1	1	**0,536**	0,523
A4	0,431	0,468	0,412	0,454	**0,536**	0,523	1	1

Table 5.5: K scores obtained on the prior (O[riginal]) and post (R[evised]) revision annotations in DATASET-3. Higher and lower values of agreement for the original and revised annotations are bolded.

	Created PRs	**Revised (%)**	**Deleted (%)**	**Modified (%)**	**Confirmed (%)**
A1	140	39 (27.86%)	11 (28.20%)	4 (10.26%)	24 (61.54%)
A2	252	85 (33.73%)	21 (24.71%)	25 (29.41%)	39 (45.88%)
A3	197	50 (25.38%)	15 (30%)	10 (20%)	25 (50%)
A4	163	46 (28.22%)	20 (43.48%)	9 (19.57%)	17(36.96%)

Table 5.6: Number of PRs created during the annotation (pre-revision) and absolute and relative number over PRs of revised pairs for each annotator. Among the revised, the table reports the absolute and relative number of deleted, modified and confirmed PRs.

	ERROR TYPES				
	Background Knowledge	**Too Far**	**Annotation Error**	**Wrong Direction**	**Co-PR**
A1	0.00%	36.36%	**63.64%**	0.00%	0.00%
A2	4.76%	**66.67%**	28.57%	0.00%	0.00%
A3	13.33%	20.00%	**53.33%**	6.67%	6.67%
A4	15.00%	30.00%	**45.00%**	0.00%	10.00%

Table 5.7: 'Error Type' columns report for each annotator the percentage of deleted pairs assigned to each label.

best modifying, the weight of their pairs. We also asked experts to express the reason that brought them to delete a pair, by choosing among five possible motivations that we identified during a previous and unreported case study conducted on DATASET-2. All motivations indicated by the four annotators of DATASET-3 felt between reasons 1 and 5: generic annotation errors due to distraction are the most common type of mistake they made, except for annotator 2 that identified "too far" as the main cause of error in his annotation. Surprisingly, annotators did not consider that their background knowledge as experts interfered with the annotation, nor that their mistakes were due to formal errors, such as wrong directions.

5.4 Automatic identification of PR relations

In section 4.2.4 we presented the extraction module and a set of methods that we selected among those proposed in the literature. In that section we also provided details about each method, their features and weak vs strong points. We implemented such methods and we made them available in the framework. In addition, we developed a new approach for prerequisite extraction. This section deals with the description of our approach for PR automatic identification.
In [3] we first proposed an approach for prerequisite relation extraction from an educational textbook, by combining two methods: (1) using temporal ordering and co-occurrence of concepts, (2) using the structure of the textbook and the relevance of the terms. In the present work we propose an alternative approach to method 1: instead of using only co-occurrence of concepts and temporal ordering, we propose the use of *burst analysis* [119] *based on co-occurrence and combined with temporal reasoning*. Burst analysis has already been used in text mining for summarization [212] and relation extraction [242]. It is based on the idea that terms in a text have bursting intervals, i.e. portions of text where they are particularly prominent. Relations between pairs of concepts are derived by observing how pairs of burst intervals that belong to different terms are positioned in the text flow.

For the experimental evaluation (see section 5.4.2) we compared our method for PR relation extraction, based on burst analysis and co-occurrence, against a set of baselines and using the revised and combined version of DATASET-3 (see tables 5.2 and 5.5). The experimental evaluation provides promising results in terms of Precision and Recall of PR identification. Moreover, we conducted two additional experiments, discussed in section 5.4.3, one focusing on a comparison between burst analysis and co-occurrence (section 5.4.3.1) and the other concerning the use of bursting intervals to feed a neural architecture (section 5.4.3.2).

The main contribution of our work on PR extraction to the literature is the expansion of prerequisite extraction through an unsupervised and domain independent approach, which exploits only the unstructured content of a digital textbook, i.e. without using external resources such as Wikipedia links [214] or other knowledge bases [171].

[130, 242] use burst analysis to recognize relationships between concepts and draw them as links in a CM. Contrary to us, method in [130, 242] extracts all the possible relations between pairs of concepts, while our effort is to identify specifically PR relations. We perform PR extraction starting from the educational material where concepts are

described since a PR relation strictly depends on the writer's communicative intent and teaching style.

Notation. We define a document D as a textual resource. Concept extraction returns a terminology T, where each of its elements is a domain-specific term t_u and $t_u \in D$. Following [242], we define a burst interval B as a slice of sentences in D where the occurrences of a term t_u are denser than in other segments and $B_{t_u}[i]$ is the *i-th* bursting period of term t_u. The final output of concepts and PR relations extraction is a concept graph G represented, similarly to [236], as a set of triples in the form $G = \{(t_u, t_v, p) | t_u, t_v \in T, p \geq 0\}$, where p is a positive value indicating the strength of the PR relation between t_u and t_v (t_u prerequisite of t_v).

5.4.1 Burst–based Method

In the present work we propose a new approach for building the concept graph. As mentioned above, co-occurrence is not always a satisfying measure of PR relation, since it often overestimates PR relations, including other kinds of relations between terms that frequently co-occur. Moreover, deciding which concept plays the role of prerequisite in a pair, by considering only their temporal order of appearance in the text, may result in a PR relation with wrong direction, where the prerequisite has been extracted as consequent and vice versa. Actually, concepts in educational textbooks may appear with different scopes along the text flow: first they might be just mentioned or introduced, then used inside their definition and later recalled to explain some new information. Therefore, by viewing the textbook as a stream of sentences, one could analyze these changes and better understand how the relation between two concepts evolves in the document.

Kleinberg formally defines and models the periods of an event along a time series (e.g., a stream of documents such as e-mails or news articles) as a two state automaton in which the event is in the first state if it has a low occurrence, but then it moves to the second state if its occurrence rises above a certain threshold, and eventually it goes back to the first state if its occurrence goes below the threshold [119]. These transitions are repeated along the entire duration of a time series and the periods in which the event remains in the second state are called *burst intervals*. If applied to a single document rather than a set, Kleinberg's algorithm can be used to detect the bursting intervals of keywords [130, 242]. Intuitively, a rising of *bursting activity* associated with a concept signals its appearance or re-appearance in the flow of the discourse, revealing that certain features, mainly the frequency of the concept in that interval, are sharply rising

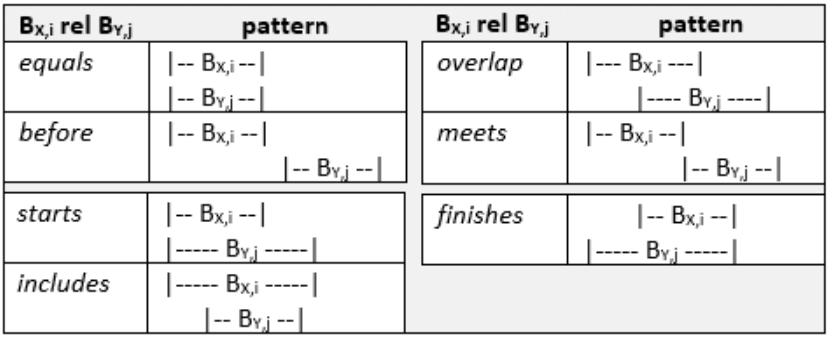

Figure 5.5: Burst Relations Interpretation

[119] and suggesting that the concept has become more prominent. With the burst detection we gain not only the intervals in which a concept t_u is “bursty" (i.e. $B_{t_u}[i]$), but also the hierarchical level of “burstiness" of these intervals. In fact, bursts associated with an event form a nested structure, with a long burst of low intensity potentially containing several shorter bursts of higher intensity inside it [119]. Moreover, the use of burst analysis also allows us to analyze different types of temporal patterns established by two concepts when they are used in the text. Our interest in applying burst analysis largely arises from the temporal nature of the PR relation: instead of using co-occurrence as criterion for the extraction, we propose to extract burst intervals of the concepts and then apply spatial-temporal reasoning on the extracted patterns in order to identify PR relations. As a matter of fact, the comparison of temporal patterns allows a richer analysis: by analyzing the pairs of intervals between two different concepts t_u and t_v, we can exploit Allen's interval algebra [8] to capture and formalize their temporal relations. Among Allen's basic relations, we used only a subset of temporal patterns, for which we could recognize some meaningful interpretation with respect to the PR relation. Consistently with our main assumption of co-occurrence as a necessary (though not sufficient) condition for a PR relation, all adopted patterns imply a co-occurrence of two terms within a temporal window. Even the *before* relation, if detected by applying a maximum gap between two intervals, entails co-occurrence. Our selection is shown in Figure 5.5. For simplicity, t_u and t_v are referred as X and Y, and $B_X[i]$ as $B_{X,i}$.

Allen's relations. Contrary to [130, 242], where combinations of burst intervals were used for identifying generic relationships between concepts, we seek to recognize the PR relation. To this aim, we make the following assumptions in order to give a prerequisite interpretation to Allen's relations.

$B_{X,i}$ ***equals*** $B_{Y,j}$ This pattern emphasizes the relatedness of two concepts without necessarily implying the existence of a PR relation. In these cases, some kind of relation between X and Y is highly probable, but we cannot say whether X is a prerequisite of Y or vice versa. Moreover, this pattern may not reveal a PR relation at all, since *equal* is very common when two concepts are co-requisites. Consequently, we assume *equal* has a low potential to reveal a prerequisite.

$B_{X,i}$ ***before*** $B_{Y,j}$ Since a prerequisite commonly precedes its subsidiary concept, in this pattern X could be probably a prerequisite of Y. We do not consider pairs of bursts with a *before* pattern when their gap exceeds a certain number of sentences, since in such cases the two concepts are almost certainly too far to establish a direct PR relation.

$B_{X,i}$ ***overlaps*** $B_{Y,j}$ If concept X is prerequisite of Y, in the text we would expect at least some cases where X is first explained and shortly after Y is introduced, with a certain area of overlapping. Thus, this pattern is highly informative for the existence of a PR relation.

$B_{X,i}$ ***meets*** $B_{Y,j}$ Here the bursting period of concept X stops exactly when concept Y begins to be more intense in the text. The two concepts are too near to completely disregard the possibility of a PR relation, and yet, as already mentioned, the proximity is not per se a sufficient condition for a PR relation. Hence, we assume this pattern has a moderate force to suggest a prerequisite.

$B_{X,i}$ ***starts*** $B_{Y,j}$ The *starts* pattern can be representative of situations where two concepts emerge almost simultaneously (most likely because they are highly related), but then the author temporarily abandons one of the two concepts while he further develops the other. According to this observation, there is a moderate/high chance that X is prerequisite of Y.

$B_{X,i}$ ***includes*** $B_{Y,j}$ This pattern shows a concept being discussed within the span of a more long-standing concept, with the longer one that totally encompasses the smaller one. The nested concept can be very likely a specification of the embedding concept at a more fine-grained level (and thus a PR relation can be appropriately traced), or sometimes it could represent a detour from the main line of discussion (and thus disclosing a learning content that is suggested for a deeper analysis). For these reasons, *includes* is highly informative.

$B_{X,i}$ ***finishes*** $B_{Y,j}$ Compared to other patterns, here a PR between X and Y is harder to assume, since $B_{Y,j} \prec B_{X,i}$. Nevertheless, a low weight should be still considered to deal with cases of bottom-up explanations.

Algorithm Description. The algorithm is structured in three phases (the pseudocode can be seen in Algorithm 1): a burst extraction phase (*ExtractBursts*), a temporal pattern detection phase (*DetectTemporalPatterns*) and a prerequisite extraction phase (*ExtractPrereqs*). The burst extraction phase (see Fig. 5.6) takes as inputs a document D containing the full text to analyze, a terminology T consisting in a list of terms appearing in D and a set of parameters for constructing the Markov's chain according to Kleinberg's description (the base s of the exponential distribution used for modeling the event frequencies, the coefficient γ for the transition costs between states, and the desidered level l within the hierarchy of the extraced intervals). D is transformed into an ordered list of sentences by means of sentence splitting, and the result is $\mathscr{Q}_D = \{q_1, q_2, ..., q_i\}$, where q_i is the i-th sentence of D. A dictionary $\mathscr{O}$ is built for mapping each concept in T with the indexes of sentences where it occurs. Burst intervals are identified for every concept t given its list $\mathscr{O}_{t_u}$ of sentence indexes, e.g. the burst intervals of t_u are: $B_{t_u} = \{[b_{starts_1} - b_{ends_1}], [b_{starts_2} - b_{ends_2}], ..., [b_{starts_i} - b_{ends_i}]\}$. The function $kleinberg(\mathscr{O}[t], s, \gamma, l)$ involves the construction of an infinite hidden Markov model as described in [119]. For this particular procedure we relied on an implementation of Kleinberg's algorithm available for Python[2] that needs to be fed with $\mathscr{O}$[t]. In addition, two parameters, s and γ, need to be set in advance: the former controls the exponential distribution from which an event is assumed to be drawn (i.e. how frequent an event must be in order to trigger the detection of a burst); the latter modifies the transition cost to a higher state. Higher values of s increase the strictness of the algorithm's criterion for how dramatic an increase of activity has to be in order to be considered as a burst; higher values of γ mean that a burst must be sustained over longer periods of time in order to be recognized [26]. During the tuning of these parameters we opt for minimal permitted values ($s = 1.05$, $\gamma = 0.0001$) with the aim of maximizing the extraction of bursting intervals. In the phase *DetectTemporalPatterns* (see Fig. 5.7), every pair of bursts $B_{t_u}[i]$ and $B_{t_v}[j]$ (belonging to two distinct concepts t_u and t_v) are compared, and temporal relations are identified by performing pattern matching. A weight W_r is therefore assigned to the identified Allen's relation r, according to the considerations described in Section *Allen's relations*. Similarly to [130], we also follow the idea that adding a tolerance gap is necessary in this stage. As a matter of fact, by considering only

[2]Library *pybursts*, https://pypi.org/project/pybursts/0.1.1/

the exact starting/ending/meeting point of two bursts, we can hardly find a complete match, while by adding a tolerance gap the method becomes more permissive during the identification of temporal patterns. The result of the current phase is a square matrix $\mathscr{P}$ of size $|\mathscr{B}| \times |\mathscr{B}|$, where $|\mathscr{B}|$ is the total number of extracted bursts, reporting a weight for each pair of bursts as resulted from the pattern matching procedure (the weight is zero only for bursts pairs with a distant *before* relation and for bursts pairs belonging to the same concept). In the PR extraction phase (see Fig. 5.8), the matrix obtained from the previous step is taken as a basis for constructing an undirected square matrix $\mathscr{M}$ of size $|T| \times |T|$: for each two distinct concepts t_u and t_v, all the weights associated with the burst pairs belonging to t_u and t_v are combined and normalized by means of the PR formula below, i.e. a modified version of the normalized relation weight (NRW) formula described in [130]. The resulting weight is stored both in $\mathscr{M}_{t_u,t_v}$ and $\mathscr{M}_{t_v,t_u}$. Given $X, Y \in T$ and $X \neq Y$, we compute $PR_{X,Y}$ as the sum of the relation weights W_r assigned to the recognized Allen's patterns, then we normalize this value by taking into account the frequency f of X and Y in their respective intervals $B_{X,i}$ and $B_{Y,j}$, the total length (measured in sentences) of all bursts of X and Y, and also the number of these bursts[3]. $\mathscr{M}$ is therefore converted into a direct matrix, and the direction is given by comparing the first bursts of the concepts in the pair. A directed graph $\mathscr{G}$, with concepts as nodes and PR relations as edges, can be finally built from $\mathscr{M}$.

$$PR_{X,Y} = \sum_i \left(W_r \, \frac{f(X, B_{X,i}) \times |\mathbf{B_X}|}{\sum_i |B_{X,i}|} \, \frac{\sum\limits_{j \in rel(B_{Y,j}, B_{X,i})} f(Y, B_{Y,j}) \times |\mathbf{B_Y}|}{\sum_j |B_{Y,j}|} \right)$$

[3]Note that the current formula takes into account all the relations where an Allen's pattern is recognized, while we are working on an improved version that limits them to relations where the subsidiary concept exhibits high burstiness.

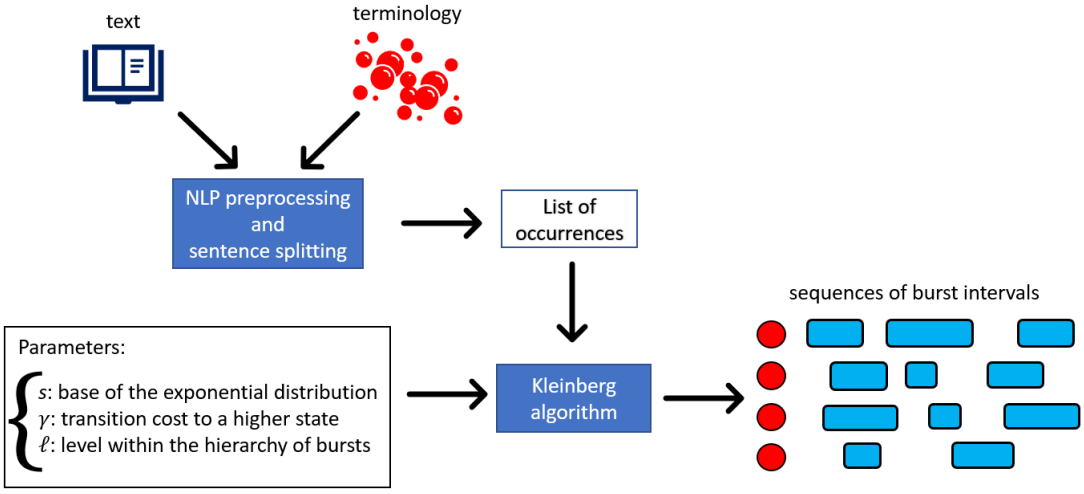

Figure 5.6: Burst Extraction Phase

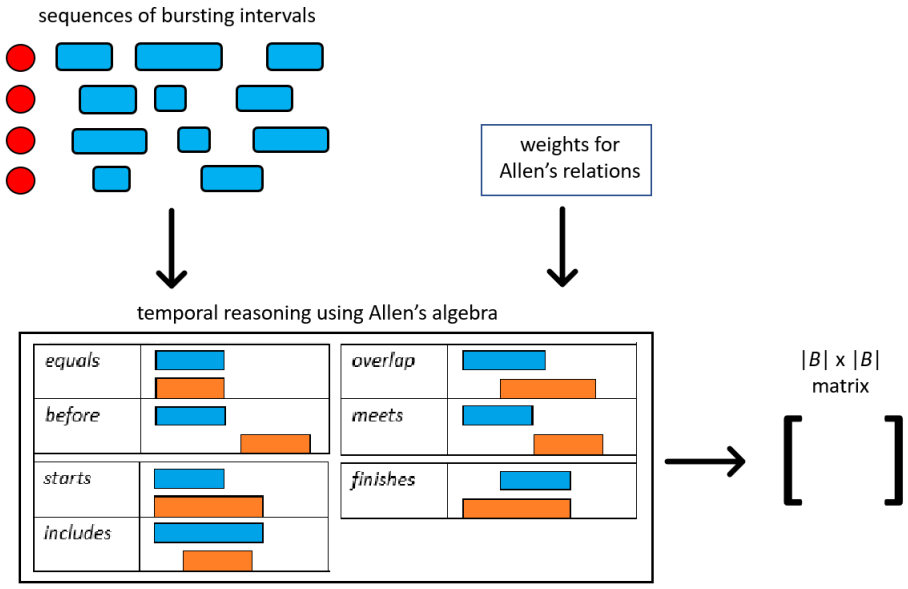

Figure 5.7: Temporal Pattern Detection Phase

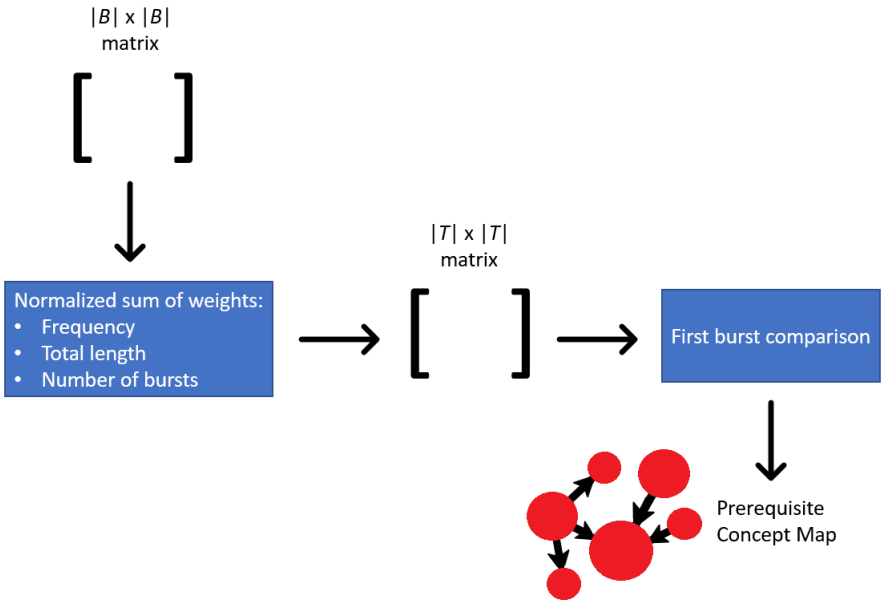

Figure 5.8: Prerequisite Extraction Phase

Algorithm 1: Burst Analysis for Prerequisite Extraction

Input: text document D; terminology T; Kleinberg's parameters (s, γ, l); dictionary of weights associated with Allen's relations $\mathcal{W}$

Output: a set of triples in the form $\mathcal{G} = \{(t_u, t_v, p) | t_u, t_v \in T, p \geq 0\}$

ExtractBursts (D, T, s, γ, l)

- **Input:** text document D; terminology T, Kleinberg's parameters (s, γ, l)
- **Output:** a set of triples in the form $\mathcal{B} = \{(t_u, start, end) | t_u \in T, 0 \leq start, end < D.length\}$
- $\mathcal{Q}_D = \{q_1, q_2, ..., q_i\}$;
- $\mathcal{O} = \{t_1 : [\emptyset], t_2 : [\emptyset], ..., t_i : [\emptyset]\}$;
- $\mathcal{B} \leftarrow \emptyset$;
- **foreach** *term* $t \in T$ **do**
  - **foreach** $q \in \mathcal{Q}_D$ **do**
    - **if** $t \in q$ **then**
      - $\mathcal{O}[t].add(\mathcal{Q}_{\mathcal{D}}.indexOf(q))$;
  - $\mathcal{B}.add(kleinberg(\mathcal{O}[t], s, \gamma, l))$;
- **return** $\mathcal{B}$;

DetectTemporalPatterns $(\mathcal{B}, \mathcal{W})$

- **Input:** a set of triples with bursts ($\mathcal{B}$); a dictionary of weights associated with Allen's relations ($\mathcal{W}$)
- **Output:** a square matrix $\mathcal{P}$ of size $|\mathcal{B}| \times |\mathcal{B}|$
- $\mathcal{P} \leftarrow$ empty square matrix;
- **foreach** *burst* $B_{t_u}[i] \in B_{t_u}$ **do**
  - **foreach** *burst* $B_{t_v}[j] \in B_{t_v}$ **do**
    - **if** $\exists rel(B_{t_u}[i], B_{t_v}[j])$ **then**
      - $\mathcal{P}[B_{t_u}[i], B_{t_v}[j]] \leftarrow \mathcal{W}[rel]$;
- **return** $\mathcal{P}$;

ExtractPrereqs

- **Input:** terminology T, square matrix $\mathcal{P}$
- **Output:** a set of triples in the form $\mathcal{G} = \{(t_u, t_v, p) | t_u, t_v \in T, p \geq 0\}$
- $\mathcal{M} \leftarrow$ empty square matrix $|T| \times |T|$;
- $\mathcal{G} \leftarrow \emptyset$;
- **foreach** $t_u \in T$ *and* $t_v \in T$ *and* $B_{t_u} \in \mathcal{P}$ *and* $B_{t_v} \in \mathcal{P}$ **do**
  - **if** $\mathcal{P}[B_{t_u}, B_{t_v}] > 0$ **then**
    - $\mathcal{M}[t_u, t_v] += PR(t_u, t_v)$;
  - **if** $B_{t_u}[0] < B_{t_v}[0]$ **then**
    - $t_u \prec t_v$
  - **else**
    - $t_v \prec t_u$
- **foreach** $t_u \in T$ *and* $t_v \in T$ **do**
  - $\mathcal{G}.add(t_u, t_v, \mathcal{M}[t_u, t_v])$;
- **return** $\mathcal{G}$;

5.4.2 Experimental Evaluation with PRET annotated and revised gold standard

As said in chapter 3, the goal of an annotation process is to provide (i) rich language resources that are useful for the analysis of the problem, and (ii) reliable gold standards that are useful in experimental settings such as training automatic systems and evaluating their performances. In this section we present an evaluation of the method described in 5.4 using the revised DATASET-3 as gold standard.

Evaluation method. PR relations are commonly analysed and evaluated as concept pairs, in line for instance with [12, 214, 236, 247]. We believe that such approach overlooks the holistic nature of PR annotation process, whose result is typically a directed graph where each path is an interpretation of a relation arisen from reading the whole text and should therefore be evaluated accounting for those peculiarities. The commonly adopted pairwise evaluation of PRs does not take into account the annotated graph as a whole; in particular, it misses the interdependence between concepts involved in PR paths and does not take into account the characteristic of PR relation (e.g. transitivity). More in general, other semantic annotations in addition to PR annotation (e.g. temporal relations [215], anaphora chains [179], discourse labelling [184]) present similar issues. In particular, temporal relation processing may represent an interesting ground of comparison for PR, for two reasons: (i) PR establishes an order of precedence between two entities (concepts) and shows also a transitive nature, (ii) researchers involved in both fields may encounter similar limitations when they need to evaluate the awareness of automatic systems using traditional performance metrics used in information retrieval community, e.g. precision and recall as well as their harmonic mean (i.e. F-score) [225]. A common scenario in both fields is when the annotation contains three elements A, B, C (pedagogical concepts or temporal events) such that $A < B$ and $B < C$, but the system identifies the relation $A < C$ (here we use the symbol $<$ to indicate both the temporal relation *before* and the prerequisite relation $\prec$): in such cases, traditional evaluation metrics will fail to identify $A < C$ as a correct relation, even if this is an implicit consequence of the other two [225]. In some cases a sort of transitive closure may therefore seem necessary in order to properly compare two graphs. In Allen's algebra [8], temporal closure is defined as a reasoning mechanism that derives implicit relations from a knowledge base where such relations are not explicitly expressed. For example, if we know that A *before* B, and B *before* C, then using temporal closure we can derive A *before* C. Temporal closure has been used in temporal relations processing to reward extracted relations

when they are distinct but equivalent to gold relations [198, 225].

Similarly, because of its characteristics, PR encourages the use of evaluation methods based on metrics that reward, or at least do not penalise, extracted relations that are not explicitly included in the expert graph but can be legitimately derived from it. The fact that they were not included by the experts does not mean indeed that they were not correct in their opinion. Even when PR annotation is performed as a concept pair labeling task, experts arguably take into account more complex patterns, and they introduce relations having in mind a larger vision of the sub-graph that they are progressively creating. A consistent evaluation approach requires to deal at least with transitive edges and path similarity between the two graphs (the experts' and extracted). In particular, for the present study, a relation between two concepts is evaluated as correctly detected when there is a path between those two concepts according to the experts.

Corpus, terminology and dataset. As evaluation dataset we used the revised version of DATASET-3 (see table 5.2 for a summary of quantitative analysis). We recall here that this dataset was obtained by asking 4 experts to annotate chapter 4 ("Networking and the Internet") of a computer science textbook [37] (20,378 tokens, distributed over 751 sentences). Before the annotation, concept extraction was addressed as a semi-automatic task by relying first on Text-To-Knowledge platform [69] for the extaction, and then asking three experts to manually revise the set of extracted terms. The final terminology T consists of 140 terms, for a total of 19460 pairs of distinct terms (representing the candidate PRs excluding symmetric pairs, i.e. $n \times n - n$ with $n = |T|$). Annotators were provided with such terminology and they annotated PRs according to the annotation methodology described in 5.1.2.2; later they revised their annotation according to the revision protocol described in 5.3. The gold dataset was finally created by combining all the revised annotations and considering as positive pairs (i.e. showing a prerequisite relation) all pairs of concepts annotated by at least one expert after the revision. The combination of all four revised annotations produced a gold standard composed of 350 unique PR relations (1.8% over all possible candidate pairs). This value is consistent with post-annotation analyses (see section 5.2), which show that PRs annotated datasets tend to be sparse, i.e. only a small subset of $n \times n - n$ possible pairs is annotated as PR by at least one annotator. Obviously, PR relations in the final gold dataset obtained different raw agreement value, i.e. they were identified by a different number of experts.

PR Relation Extraction. To identify PRs in the corpus we ran the following methods included in PRET framework (see section 4.2.4): (i) semantic relations, namely hypernymy (is–a) and meronymy (i.e. part–of), identified by relying on WordNet (we extracted

parent concepts that are located at one level higher as well as all the concepts that can be found by performing a full traversal of the hierarchical semantic tree until the root node is reached); (ii) identification in the text of lexical-syntactic patterns that may reveal the existence of a prerequisite relation (in particular, we used a selection of patterns presented in [236]); (iii) use of textbook structure (i.e. table of contents or TOC) as an indicator of prerequisite relations between concepts across sections (for the present evaluation we opted for an implementation of the metric defined in [236], called *TOC Distance*); (iv) BM, as described in 5.4.1; (v) method based on co-occurrence + temporal order, that identifies a prerequisite relation $A \prec B$ when two terms co-occur at least once in a three sentences span and the first occurrence of A in the text is before the first occurrence of B.

Results and Discussion. Results are reported in table 5.8 and are expressed in terms of precision, recall and F1 score. The table also reports the percentage of extracted relations over all possible candidates. In the remainder of the present section we discuss these outcomes.

Results of WordNet-based hyponymy/meronymy method show a high precision (0.80) but a low recall (0.01), due to a high number of false negatives. This suggests that relying exclusively on an external lexical database was not a satisfying strategy for this particular task. The reason behind such outcomes may lie in the external nature of lexical resources from which we can draw semantic relations. In fact, external lexical resources are not strictly built from the text or domain under examination during an evaluation. As a result, relations may not be extracted since the resource does not properly cover the domain of the document, and only a small set of correct relations is identified. Furthermore, as emerged during the contextual analyses of annotations (see section 5.2), hypernymy and meronymy do not fully cover the entire spectrum of PRs, since also more complex or non-hierarchical relations can be useful to detect the existence of a prerequisite pairs. As said in 2.3.1.1, in other cases lexical external resources may also lead to the opposite problem, i.e. extracted hypernym-hyponym relations can be lexically valid but not really expressed in the text. When we run the same method traversing the entire hierarchical semantic structure until the root node, the performance slightly improves, in terms of both precision and recall. This gives us an idea of the proportion of the upward tree traversal that we may need to do for finding PRs. In particular, it may be not enough searching a concept prerequisites only in its immediate superordinate (i.e. parent) concept.

Unlike lexical resources, syntactic patterns extract relations directly from a specific text by means of pattern matching, without using external knowledge bases. A typical problem with patterns though is that words must appear exactly with the predefined configuration, otherwise a relation will likely not be captured [186]. This expectation is in line with what we obtained, i.e. the low recall rate reported in table (0.01). Unfortunately, PRs in texts are not always expressed with explicit formulations such as "A is a B" or similar. Compared to hyponymy/meronymy method, lexic-syntactic patterns provided a lower precision, since also the number of false negatives is high. This may be understandable if we consider that, as we also reported in the code book (see Appendix A), syntax patterns are often also found between nominal structures that do not represent concepts (e.g. generic terms). On the other hand, this method shows a value of precision (0.60) that is comparable with respect to BM and co-occurrence and is even higher than TOC.

Unlike semantic relations and syntactic patterns, results of TOC distance show a higher recall (0.77) and a lower precision (0.14). The recall value supports the idea that tables of contents implicitly reveal some clues about prerequisite dependencies between concepts across the sections. A reasonable assumption is indeed that concepts in textbooks are distributed across sections according to some order of precedence based on pedagogical choices (e.g. from general to specific, or from basic to advanced), hence TOC can be used to extract PRs (see section 2.3.2.2 for automatic methods that reflect this idea). The dependencies inferable from a TOC are typically at a high level of granularity (i.e. chapter or section-level rather than paragraph or sentence level), therefore this source of information conveys cues for tracing inter-section rather than intra-section dependencies between concepts. The inter-section principle of the method leads to take as prerequisites all concepts occurring in a previous section. The lower precision comes from the fact that this method extracted an over-connected graph affected by false positive relations. 32.7% among all possible pairs have been indeed classified as PRs (against the percentage of 1.8% shown in the gold standard). Compared to all the previous methods, BM performs better in terms of recall (0.88) and F1 score (0.71), while precision (0.60) seems to have further room for improvement. Interestingly, the PR graph extracted by this method presents a number of relations equal to 5.7% of all possible candidate PR pairs. On the one hand, this value is still a relatively high number with respect to the gold standard, but on the other hand it constitutes a more acceptable compromise between the sparsity of the first three methods and the hyper-connectivity of TOC.

In terms of F1 and recall, co-occurrence outperforms all other methods. Albeit simple

	Hypo/Mero (until 1 level)	**Hypo/Mero (until root)**	**Syntax patterns**	**TOC Distance**	**BM**	**co-occ**
precision	0.80	**0.90**	0.60	0.14	0.60	0.58
recall	0.01	0.03	0.01	0.77	0.88	**0.95**
F1	0.02	0.05	0.02	0.23	0.71	**0.72**
% extracted PRs	0.03	0.05	0.03	32.69	5.69	7.16

Table 5.8: BM evaluation against baselines considering paths in the graph.

in its nature and not very precise (0.58), this method suggests therefore that a combination of co-occurrence and temporal order may quite well constitute an efficient strategy to capture PRs, at least with our settings. Remember in fact that in our implementation co-occurrences are captured considering a window of context made up of 3 sentences, and the entire text is lemmatized (hence an occurrence can be found even when the concept appears with an inflected form such as a plural). More in detail, the strength of this method appears to be its recall (0.95), while for F1 it behaves only slightly better than BM (0.72 against 0.71) and its precision is roughly comparable to those of BM and syntax patterns (0.58 ~ 0.60).

5.4.3 Additional evaluations

In the present section we report two additional experimental evaluations that we conducted on PR automatic extraction. The first (section 5.4.3.1) concerns a comparison between the burst–based method described in 5.4.1 and a co-occurrence based method using an expert manual validation of the extracted relations. The second (section 5.4.3.2) involves the use of burst intervals to feed a neural architecture trained for learning units ordering.

5.4.3.1 Comparison of BM with co-occurrence

For the present experimental evaluation we compared our BM method for PR relation extraction (see section 5.4.1), based on burst analysis and co-occurrence, against a baseline method based exclusively on co-occurrence. Co-occurrence based methods are the core of many approaches for PR relation identification [99, 135]. As emerged in our experiment reported in 5.4.2, co-occurrence is an intuitive condition for PR. However, a high value of co-occurrence is not necessarily a measure of PR strength, since it could identify other types of relations, such as associations, taxonomic relations and co-requisites among others. Therefore we asked domain experts to manually annotate the relations automati-

cally extracted by both methods and compared the results. The experimental evaluation provides promising results in terms of Accuracy of PR identification and Precision of top identified relations. In particular, preliminary results suggest the effectiveness of burst analysis for filtering out relationships between concepts that co-occur frequently but are not relevant for the educational purpose in terms of PR relations.

Goal. The method we proposed for PR relation identification is based on the assumption that co-occurrence of concepts is likely a condition for the existence of PR relation between two concepts. In general, high co-occurence frequency is a good indicator of relations (as shown in previous works [60, 135]), thus it can also underpin other kinds of relations besides PR. The goal of our evaluation is to investigate the following two hypotheses: (**HP1**) burst analysis, as in our proposed method, could perform better than methods based only on co-occurrence frequency, reducing false positive and false negative PR relations; (**HP2**) burst-based method for PR identification could reduce false positive PR relations when two high co-occurring concepts are related by a relation that is not PR.

Methodology. We tested our Burst-based method on a chapter of a computer science textbook, "*Computer Science: An Overview*" [37]. The output of the algorithm is compared against a method based on co-occurrence of terms in a window of context. Both methods are manually evaluated and compared by domain experts. In the following we call such methods respectively Burst-based method (**BM**) and Frequency-based method (**FM**). To test HP1, we computed the Accuracy [168] of BM and FM on a set of 150 randomly selected relations from the results of BM and FM. To test HP2, we computed the Precision [168] of the Top 150 PR relations returned by the algorithms and therefore analyzed the types of error. Details are in the following.

Corpus and Concept Extraction. For the evaluation we used chapter 4 "Networking and the Internet" of the above mentioned textbook [37] (20,378 tokens, distributed over 751 sentences). Concept extraction is addressed by relying on Text-To-Knowledge platform [69]. The extracted terminology contained both single nominal structures (e.g. *computer*) and complex nominal structures with modifiers (e.g. *hypertext transfer protocol*). The set of extracted terms was manually revised by three experts and missing terms were added. The final terminology consists of 125 terms, for a total of 15,500 pairs of distinct terms (representing the candidate PRs), excluding symmetric pairs.

PR Relation Extraction. We ran the burst algorithm as described above, assigning W_r weights to burst relations (according to the assumptions discussed in Section 5.4.1) on a 10 point scale, and thus we obtained a direct matrix. On the other hand, the FM method computes how many times two terms of the terminology appear together in a

three sentences span (i.e. the one where a term appears, the preceding and the following). The output of FM is a direct matrix as well: values represent co-occurrence frequencies of each term pairs and the direction is given by the order of first occurrence of the terms.

Experts' Annotation. In order to evaluate **HP1** and **HP2**, we asked domain experts to annotate the extracted pairs of concepts, in line with [78, 99, 135, 136]. For the first hypothesis (**HP1**), we created a sample by randomly selecting, for each method, 150 pairs of concepts using the following criteria (see Fig. 5.9): (1) 50 pairs identified by the method as having PR relationship, (2) 50 pairs identified by the method as not having PR relation, (3) 50 pairs (among those not selected for the other partitions) regardless of whether they have been identified with a PR relation or not. The third set was done to make the sample more homogeneous with respect to the algorithms' outputs, which are significantly unbalanced (i.e. only 5.11% of the pairs obtained a PR label for the BM method and 4.07% for the FM method). To evaluate the second hypothesis (**HP2**), we selected the Top 150 relations returned by each method, ordered according to their weight (see Fig. 5.10).
In both cases (**HP1** and **HP2** evaluations), two domain experts were asked to annotate the pairs in the two samples, assigning what they believed to be the correct label for that pair. The guidelines for evaluation explicitly dictated to read the textbook and assign labels based on how concepts are addressed in the text. Moreover, a third expert was asked to analyze cases of disagreement between annotators in order to check if disagreement was due to annotators' subjectivity or to annotation errors (e.g., distraction, misinterpretation of guidelines or misinterpretation of the text). The risk of errors is well-known in the literature as well as the disagreement due to annotators' subjectivity [12, 78, 85].

Results and Discussion. In order to test **HP1** (i.e. if our BM method could produce less false positives and false negatives compared to co-occurrence-only based methods), we computed the Accuracy, as defined in [168], of BM and FM. To this aim, we compared the output of the algorithms for the 150 randomly selected pairs of concepts against each expert's annotation, and then we took their average score. Notice that in this evaluation our aim was to assess the correctness of the PR relation identification, not its strength. Results in **Fig.5.11** show that BM Accuracy is 0.84, slightly performing better than FM, whose Accuracy is 0.77.
Considering **HP2** (i.e. BM could reduce false positives in cases of frequently co-occurring concepts connected by a relation that is not a PR relation), we took into account the Top 150 relations returned by each method. Obviously, for the FM method such relations

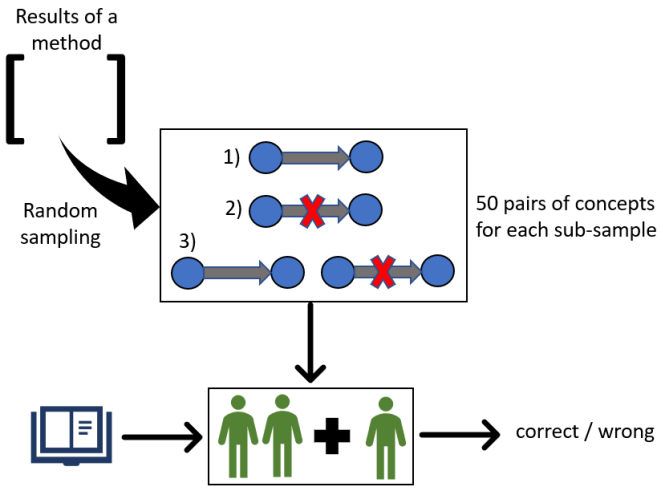

Figure 5.9: Evaluation 1

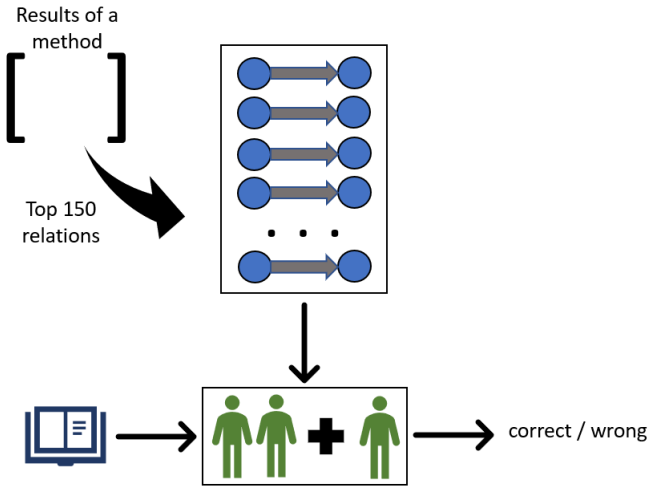

Figure 5.10: Evaluation 2

Accuracy (tp+tn)/ (tp+tn+fp+fn)	BM		FM	
	Expert 1	Expert 2	Expert 1	Expert 2
	0.82	0.86	0.8	0.75
Avg accur	0.84		0.77	

Figure 3(a)

Errors 85 Top PR	BM		FM	
	Errors	%	Errors	%
(a) distant relation	9	0.27	8	0.24
(b) Rel <> PR	4	0.12	14	0.41
(c) inverse relation	10	0.30	7	0.21
(d) no relation	10	0.30	5	0.15

Figure 3(c)

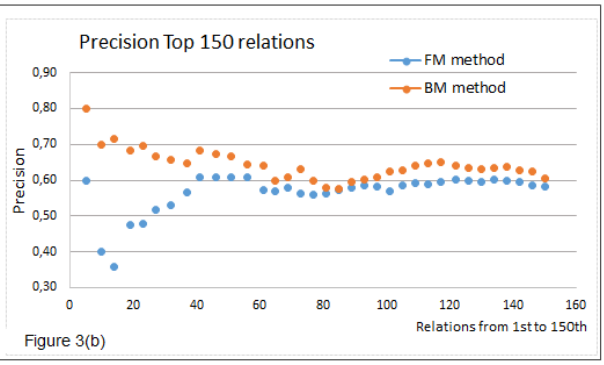

Figure 3(b)

Figure 5.11: Results: (a) Accuracy, (b) Precision top 150 PR, (c)Errors top 85 PR

are those whose concept pairs have the highest co-occurrence frequency. First of all, we computed Precision (as defined in [168]) of both methods against the experts' annotations. As displayed in **Fig.5.11**, Precision of BM performs better than FM with a decreasing trend, suggesting that BM method works better especially in cases of high co-occurrence frequency. To deepen this analysis, we also performed a qualitative analysis on the error types occurring in the top 85 relations (as shown in the chart, at point 85 we have the minimum distance between BM and FM Precision). The third expert was asked to classify the errors as: (a) very distant relation, (b) relation different from PR, (c) inverse relation, (d) no relation. As can be seen in **Fig.5.11**, *relations different from PR* are 12% with BM method and 41% with FM method, confirming that BM method allows to reduce the identification of non-PR relations between high co-occurring concepts. These preliminary results seem promising if we also consider that, by tuning the weights of Burst relations, we could further improve the outcomes. Despite the results, future work is already planned to test variations of the formula. As mentioned above the current formula takes into account all the relations where an Allen's pattern is recognized, but we are working on an improved version that limits them to relations where the subsidiary concept exhibits high burstiness. The next section will propose a further hypothesis of improvement that came out by using the visualization methods for PR relations. Moreover, the last chapter of the thesis will discuss other improvements both for the algorithm and for the evaluation.

5.4.3.2 Burst intervals as input for a neural architecture

Datasets produced within PRET framework, as well as burst intervals extracted as described in 5.4.1, were also used and tested in other experimental settings. Among them, we briefly present in this section an experiment concerning prerequisite learning classification between educational concepts. A complete description of the experiment, with details regarding methodology, experimental settings, results and discussion, is found in [13].

Classifier and burst intervals. The proposed system was developed by adapting a deep learning classification algorithm that was originally designed for sequencing Learning Objects. This system uses pre-trained word embeddings (WE) and global features automatically extracted from the data (e.g. co-occurrences of concepts, concepts frequency, text length in words, Jaccard similarity between textual contents of the two learning units, LDA, see [154] for a complete description). We applied this neural architecture to the task of ordering concepts in a textbook according to their prerequisite relations. As for the burst-based method previously described in 5.4.1, this approach identifies prerequisite relations between concepts without using any external knowledge base. For training and testing our system we relied on DATASET-2 (described in 5.1). Burst intervals were instead used to select relevant content of the textbook for each concept, i.e. to generate subsets of the textbook and give them as input to the classifier. Since the original classification algorithm was designed to receive Learning Objects as input, in order to apply the neural architecture to the new task, we automatically created for each concept in the textbook a set of sentences extracted from the text using different criteria. While the neural architecture remained the same throughout all the experiments, input textual data varied with respect to the criterion we used to generate simulated units of learning for each concept by retrieving textual content from the textbook. In doing so, we were able to study performance variations of the classifier given different input data. Given a concept, we generated subsets of the textbook according to the following criteria: (1) considering all sentences showing an occurrence of the concept (Occurrence Model); (2) considering burst intervals of each concept extracted according to the strategy described in 5.4.1 (Burst Intervals Model). Note that for this experiment we only used the bursts detected with the first phase of the algorithm described in 5.4.1, while the temporal reasoning is not employed here and we are planning to use it in the future.

5.5 Visualization of annotated and automatically extracted relations

The research presented in this section investigates the use of information visualisation techniques for better understanding and exploring prerequisite relation and its characteristics in textbooks.

Our research contributes for better understanding and exploring the phenomenon of PR in textbooks, by providing a collection of visualisation techniques for PR exploration and analysis, that we used for the design of and then the refinement of our algorithm for PR extraction. This work has been presented in [174].

Our research on PR extraction from textbooks is enhanced by the use of Information Visualisation techniques in the following phases:

(i) Exploring and discovering insights of PR;

(ii) Refining the algorithm of PR extraction by means of visual analysis of patterns and comparison between gold standard PR graphs vs extracted PR graph.

In the first phase (i), visualisation analysis techniques were applied to a concept map manually created by experts. The purpose of map creation was to make explicit the pedagogical relations among concepts in the textbook, while the aim of visualisation analysis was to discover new insights into PR. The dataset was explored through matrix and graph visualisations, both enhanced with filtering and ordering functions. This analysis supported the definition of the algorithm in 5.4.1 for PR extraction.

In the second phase (ii), visualisation analysis was applied on a map automatically extracted from a textbook using the strategy described in 5.4.1. We applied visualisation analysis with the aim of improving pattern discovery, refining the algorithm and better understanding how the automatic approach is affected by changing the parameters. In this phase we relied on a gantt representation of the algorithm results. Further analysis was conducted by “visually" comparing the extracted map and the gold map with the purpose of analysing graph differences at various levels.

In the implemented prototype of the framework, once the annotation is completed or a method output its results, the user can choose to generate different types of visualization for this data. We provide the following different views: Matrix (ordered by concept frequency, clusters, temporal, occurrence or alphabetic order), Arc Diagram, Graph and Clusters. Most of the information visualisation analysis tools that we propose are meant for the analyst (e.g., researcher) who intends to discover new insights or confirm existing

hypotheses on the PR. Nevertheless, some of these techniques/tools give a graphical visualisation that can be potentially useful also for learners, teachers or instructional designers [61, 75, 244]. For example, from this perspective a graph representation can be proposed as a supporting tool in a question answering scenario where the underneath knowledge structure is used to retrieve the most appropriate learning path without leaving out prerequisite concepts [5]. Such a tool can produce a graphical representation that reflects and explicates the necessary prerequisite knowledge or deepening knowledge in respect of the learner's query. While the latter user and teacher-centric case is left for future works, in the rest of this section we will focus on (i) and (ii).

In the following we describe our approach, techniques and data used for visualisation analysis for both the phases described above (i.e. PR exploration in Section 5.5.1 and PR extraction and algorithm refinement in Section 5.5.2), and for each phase we discuss the results.

5.5.1 PR exploration

Concept Graphs. Several variants of network-like representations (see for instance **Fig. 5.12**) have been used during PR exploration since the creation of DATASET-1, in order to visually detect elements such as loops (as resulting from human errors during the process of annotation) and transitive edges. However, as the dataset becomes larger, a concept graph becomes harder to explore, especially if no filtering functions are implemented. In this case, other forms of visualisation are more effective.

Concept Matrix Chart. This is a dynamic and interactive representation of a $|T| \times |T|$ asymmetric adjacency matrix M, where each colored cell $M_{i,j}$ represents a prerequisite relation between concepts i and j (see **Fig. 5.13**). Different colors can help to visually differentiate clusters of concepts, as they have been recognised by a community detection algorithm [4]. Intuitively these clusters shows the membership of a concept within a thematic unit (e.g., concepts related to network security, or to network classification, and so on). Different shades of the same color can be used to encode different degrees of inter-agreement among annotators (if M is used to visually depict a gold standard) or different scores (if M represents the output of an automatic method). The matrix arrangement is dynamic, i.e. the concepts along the matrix can be sorted according to different criteria: order of first appearance in the text, alphabetical order, frequency and cluster membership.

[4] In our implementation depicted in **Fig.5.13** we used the Infomap algorithm.

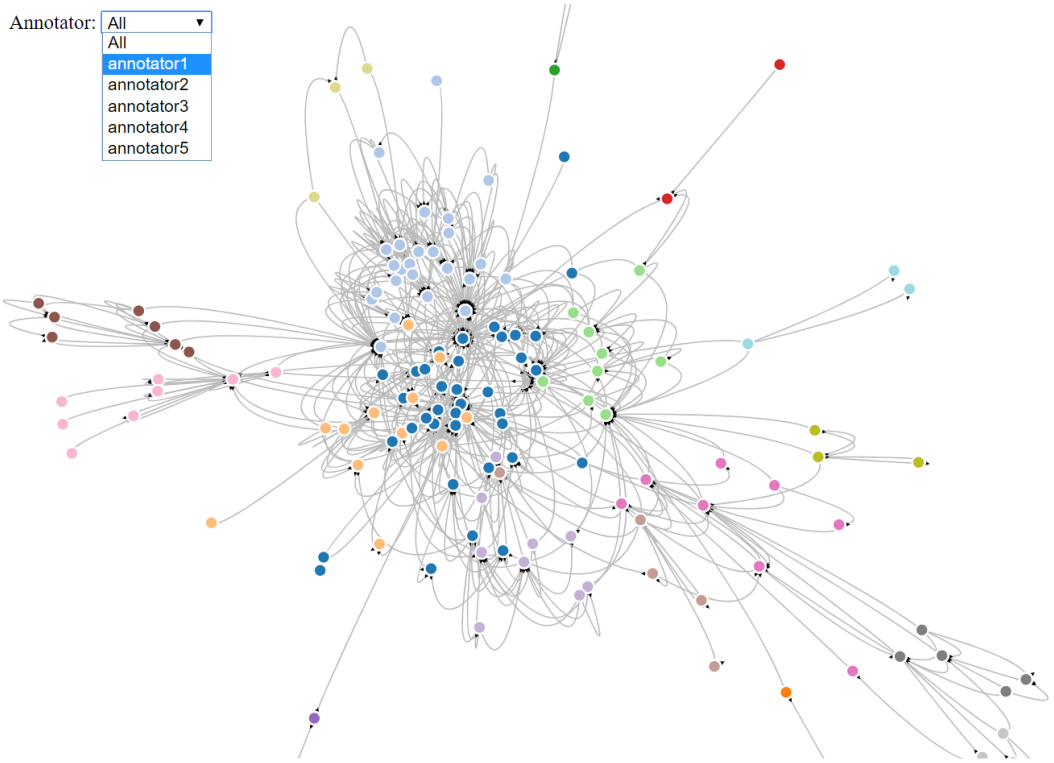

Figure 5.12: A Concept Graph that allows decomposition in sub-graphs belonging to individual annotators.

Discussions and Results. The analysis performed on the concept graph built from the gold standard allowed to reveal interesting properties concerning graph's transitivity, topology and connectivity. By comparing subgraphs belonging to different experts, we discovered that the number of transitive edges largely varies from annotator to annotator. Thanks to this observation, we discussed the phenomenon with the annotators, ascertaining that their choices depend on a different interpretation given to the meaning of a distant or weak prerequisite relation. Some of the experts tend to think in terms of graph paths, while others in terms of didactic sequences. As an example, the relation between "computer" and "local area network" (LAN) can be seen on the one hand as a transitive relation (if one has in mind the path in the graph connecting the first concept to the second by means of several bridging concepts in the middle), but on the other hand it can also be seen as a direct prerequisite relation (if one realises that "computer" is a fundamental notion, without which a student cannot possibly hope to understand what a LAN is). Concerning the topology, graph visualisation confirmed our intuition that prerequisite relations do not necessarily replicate ontological relations. As an example, let us take a

pair of concepts such as “client side" and “server side": in a domain ontology these would very probably be represented as sibling nodes, but we cannot always expect the same behaviour when approaching a didactic text. In similar contexts, even if a co-requisite relation would seem the most natural choice of presenting these kind of concepts (i.e. the author explains them together, and the former is not a prerequisite of the latter, nor vice versa), a prerequisite relation is still possible (e.g., if the author first explains the former and then relies on the knowledge gained by the reader to explain the latter). As can be noted in the graph, hypernym-hyponym and holonym-meronym relations deserve a similar discussion. External lexical resources would typically categorise pairs of words such as “device" (broader, hence at top-level) and “hub" (narrower, hence at bottom-level) or “byte" (the whole) and “bit" (the part of) in a hierarchical manner. Conversely, in textbooks (sometimes even in the same textbook) we can easily find both top-down and bottom-up explanations. Lastly, the connectivity of the graph (which we discussed above as influenced by the annotators' perception of what a prerequisite relation is) also largely depends on the annotator's level of domain knowledge.

The analysis performed on the Concept Matrix Chart built from the gold dataset revealed an important insight for the direction of the prerequisite relation. After applying the first sorting criterion (i.e. order of first appearance), the matrix tends towards an upper triangular, with colored cells mostly concentrated in the area that is slightly above the diagonal. This pattern confirms the hypothesis that prerequisite relation is highly correlated with co-occurrence and temporal order. Consequently, the temporal order of concepts is a reliable criterion to assign a direction to relations that are automatically extracted by an algorithm. The most notable exception in this pattern is represented by concepts such as “computer" or “network", which tend to be spread across the entire row of the matrix. However, this phenomenon is due to the fact that these are the main concepts of the whole chapter of the textbook, hence they frequently re-occur along the entire text and moreover they could commonly be prerequisites (rather than subsidiaries) of many other concepts.

The analyses above supported the definition of the algorithm presented in 5.4.1 for PR extraction, which is based on Burst analysis and temporal order.

5.5.2 Algorithm Refinement

Burst Dataset The method devised to obtain the Burst Map dataset exploits burst analysis [119] based on co-occurrence of relevant terms in a text and combined with temporal ordering, as described in 5.4.1. Burst analysis is based on the observation

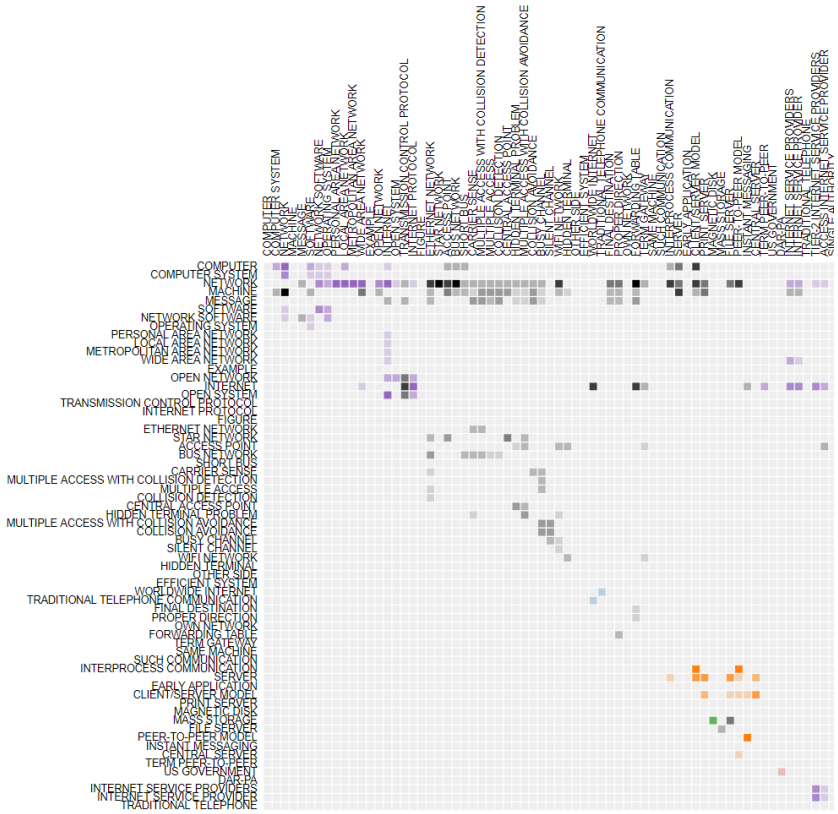

Figure 5.13: Concept Matrix Chart

of *burst intervals* of a phenomenon, that is the periods of time when the phenomenon is particularly relevant along a time series (i.e. its occurrence rises above a certain threshold) [119]. Following [130, 242], we applied burst analysis to detect the bursting intervals of relevant terms along the textbook chapter and analyse different types of temporal patterns established by two concepts by applying spatial-temporal reasoning on the extracted patterns in order to identify PR relations. To capture and formalise their temporal relations, we exploited a subset of temporal relation defined by Allen's interval algebra [8]. Our selection is shown **Fig. 5.5**. The result of the process is a concept graph with 353 concept nodes and 124,256 possible pairs of distinct concepts related by a PR.

Burst Gantt Chart. This is a Gantt diagram showing bursts of concepts along the horizontal temporal axis (time can be measured in sentences or tokens), while concepts are arranged along the vertical axis, according to their temporal order (see **Fig. 5.14**). The main purpose of this visualisation is facilitating the analysis of temporal patterns between intervals of different concepts.

Moreover, as the chart incorporates data taken from three different sources (the

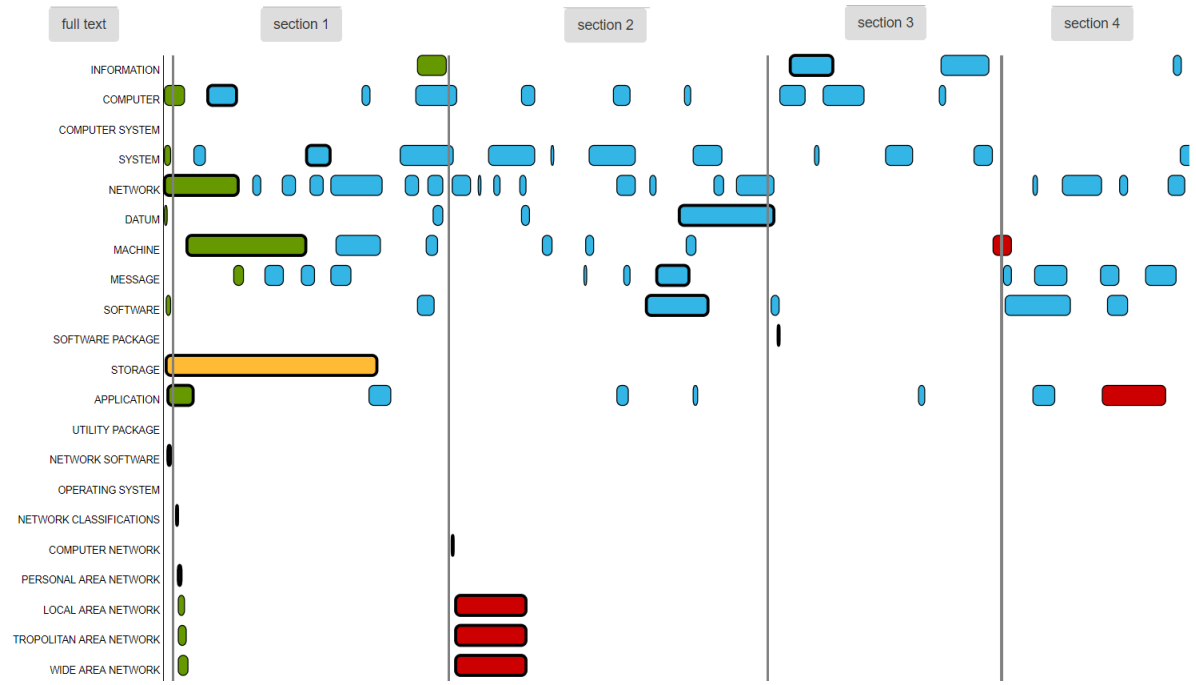

Figure 5.14: Burst Gantt Chart

output of the burst algorithm, the gold dataset and the textbook itself), we can use it to perform further kinds of analysis and textbook exploration. For instance, by clicking on a concept label in the vertical axis, we can compare Allen's temporal relations and gold relations and thus investigate possible matches (see **Fig. 5.15**).

By clicking on a burst we can instead read the portion of the textbook covered by that interval (see **Fig. 5.16**). This procedure enables us to easily find blocks of sentences where a concept is introduced for the first time, then resumed (with or without another concept) and eventually left behind. Vertical partition lines have been drawn to indicate boundaries between sections, while other sorts of markers can be traced near the temporal axis to identify sequences of sentences that according to experts are particularly rich of prerequisite relations.

Concept Graph with Allen's Relations. For investigating Allen's temporal patterns and prerequisite relations, we also propose to transform the Burst Gantt Chart into a weighted directed and edge-labeled graph G_A, where edges are labelled using Allen's algebra. For each two distinct concepts X and Y in the Burst Gantt Chart, if a pair of bursts $B_{x,i}$ and $B_{y,j}$ is related n times by Allen's relation a, we represent this configuration in G_A as $X \rightarrow_{(a,n)} Y$, where (a,n) are the edge label and edge weight

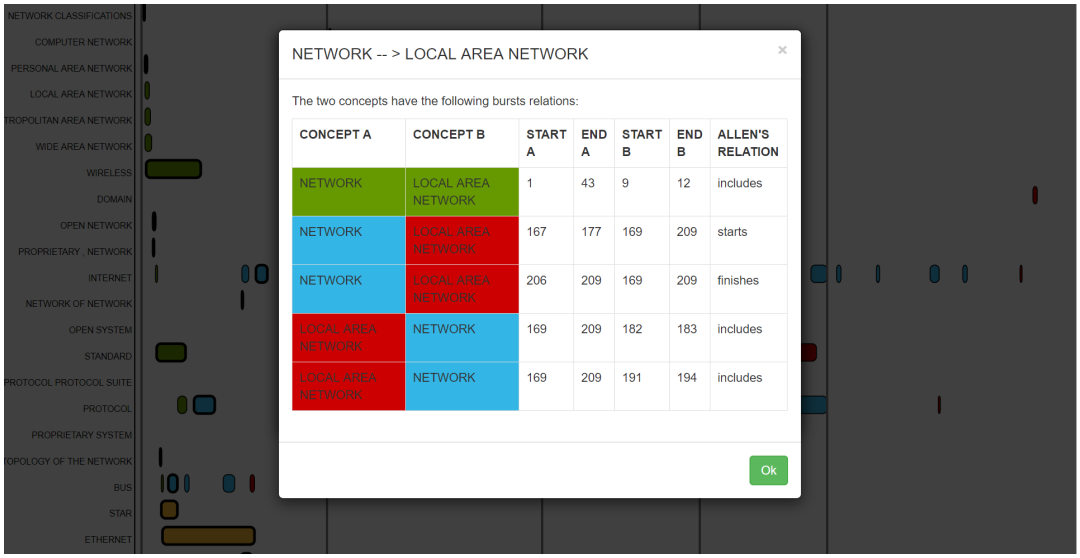

Figure 5.15: Analysis of temporal patterns.

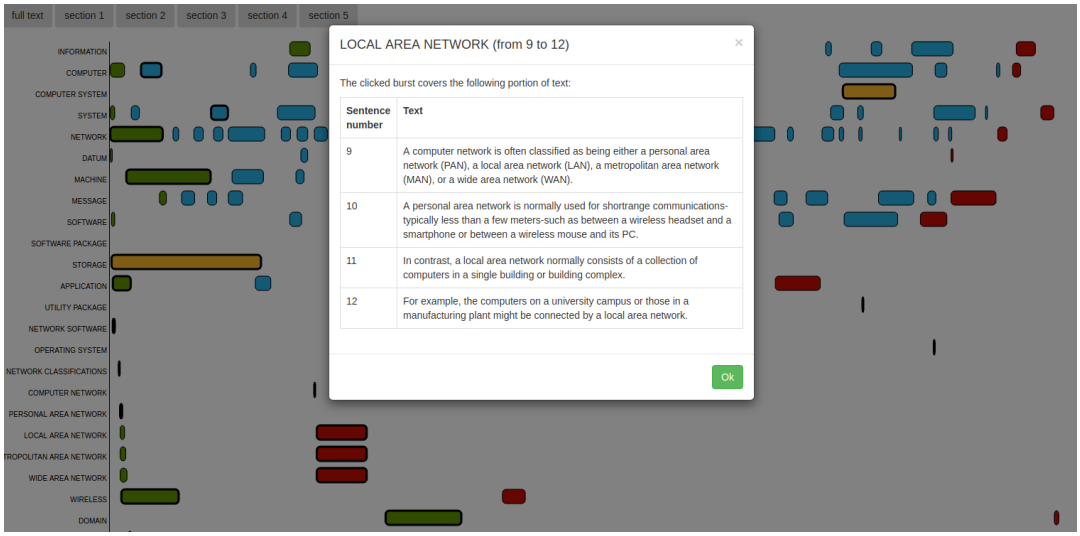

Figure 5.16: Textbook Exploration with Gantt

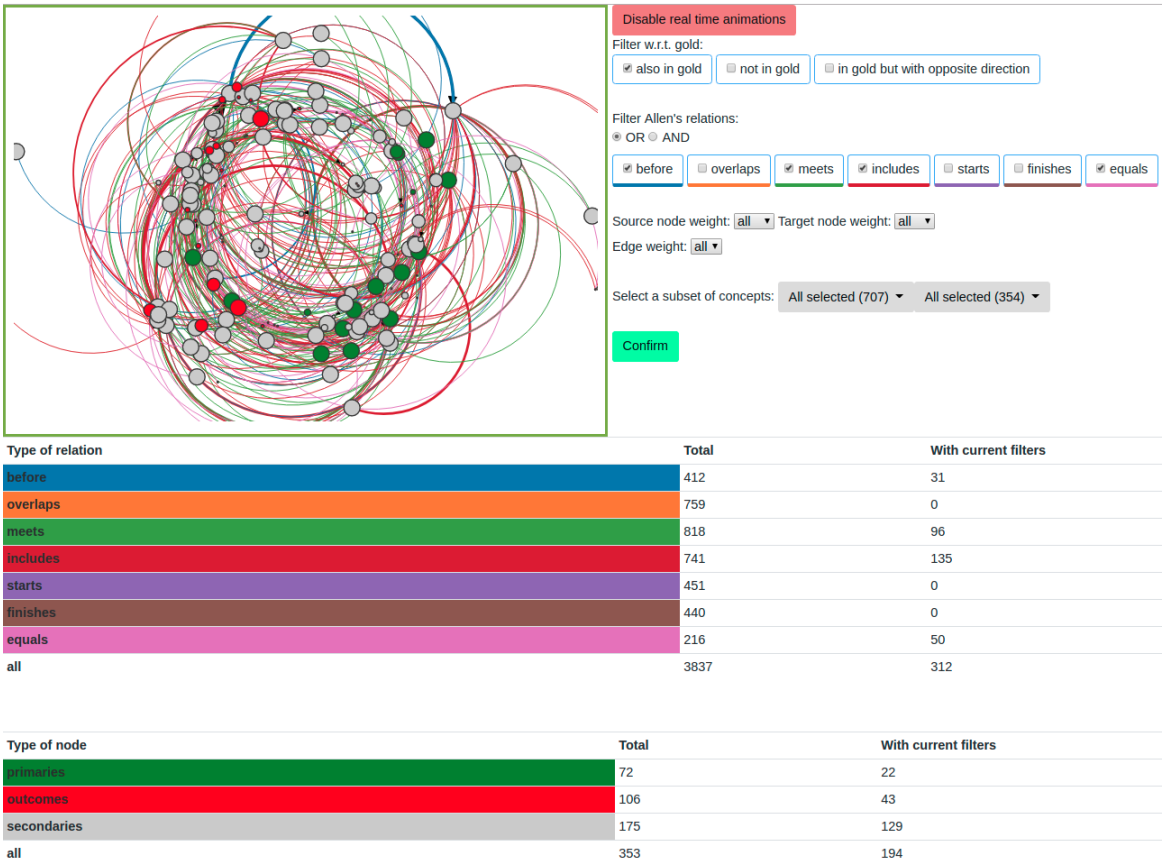

Figure 5.17: Concept Graph with Allen's Relations

respectively.

In this representation, bursts are collapsed into one node for each concept, while multiple edges are maintained and a weight is assigned to them according to how many times that temporal relation occurs between bursts of those two concepts. The aim of this conversion is producing a graph that can be compared with the gold standard graph, as a means for achieving more confidence on the weights that should be assigned to the different Allen's patterns. The result of the conversion is a highly connected graph which needs to be explored with filters, e.g., filtering concepts that have different relevance, or filtering by specific Allen's relation or combination of relations, as well as filtering according to edges or nodes weights. As displayed in **Fig. 5.17**, different colors for edges show different Allen's relations, while the width is proportional to the number of times an Allen's relation is founded between two concepts. The dimension of a node is proportional

to the importance of the concept (this value can be measured using frequency, relevance or summing all the lengths of its bursting periods in the text). In our implementation we also used different colors to encode concepts that in the gold standard are sources, sinks or internal nodes. In the first case the node has zero indegree and this means that for annotators it represents a primary notion—a concept already known by the learner; in the second case the node has zero outdegree and thus it may be intended as a final learning outcome.

Discussions and Results. Allen's patterns allow to capture PR relations quite well (see 5.4.1), however they overestimate the PR relations. This comes as a straightforward observation considering the Concept Allen Graph visualisation. As it can be seen, the number of detected Allen's relations is much bigger than the set of relations identified by the experts (even when transitive closure is applied on the experts' graph in order to reduce variety in the number of transitive edges).

Therefore, the Burst Gantt Chart was used to analyse possible combinations of Allen's patterns and more sophisticated conditions that should be satisfied between bursts of two concepts. As a result of the aforementioned analyses, we observed that Allen's Algebra, as used in our Burst-based algorithm, is likely to fail when an Allen relation is identified between bursts of concepts X and Y, but no bursts of X are present in the text before that relation. This is consistent with the intuitive consideration that concept X should be introduced before Y in order to be a prerequisite of Y, and thus for two concepts X and Y, a necessary but not sufficient condition in order to have X prerequisite of Y is that X should be previously explained, i.e. $|B_X| > 1$. Considering bursts instead of simple occurrences of a term allows to exclude cases where X occurs before Y and X is not really explained but rather simply introduced (the analysis of the text showed for example several cases where the content of the next section is mentioned before, as a guide for the reader).

As future work, we plan to implement refinements of the algorithm that take this into account. This is not trivial, since for instance the condition $|B_X| > 1$ does not apply in cases where X is a primary notion, namely a concept already known by the learner as background knowledge.

Furthermore, we plan to use Concept Allen Graph to explore combinations of Allen's patterns by filtering them in conjunction or disjunction and comparing the results with the gold standard.

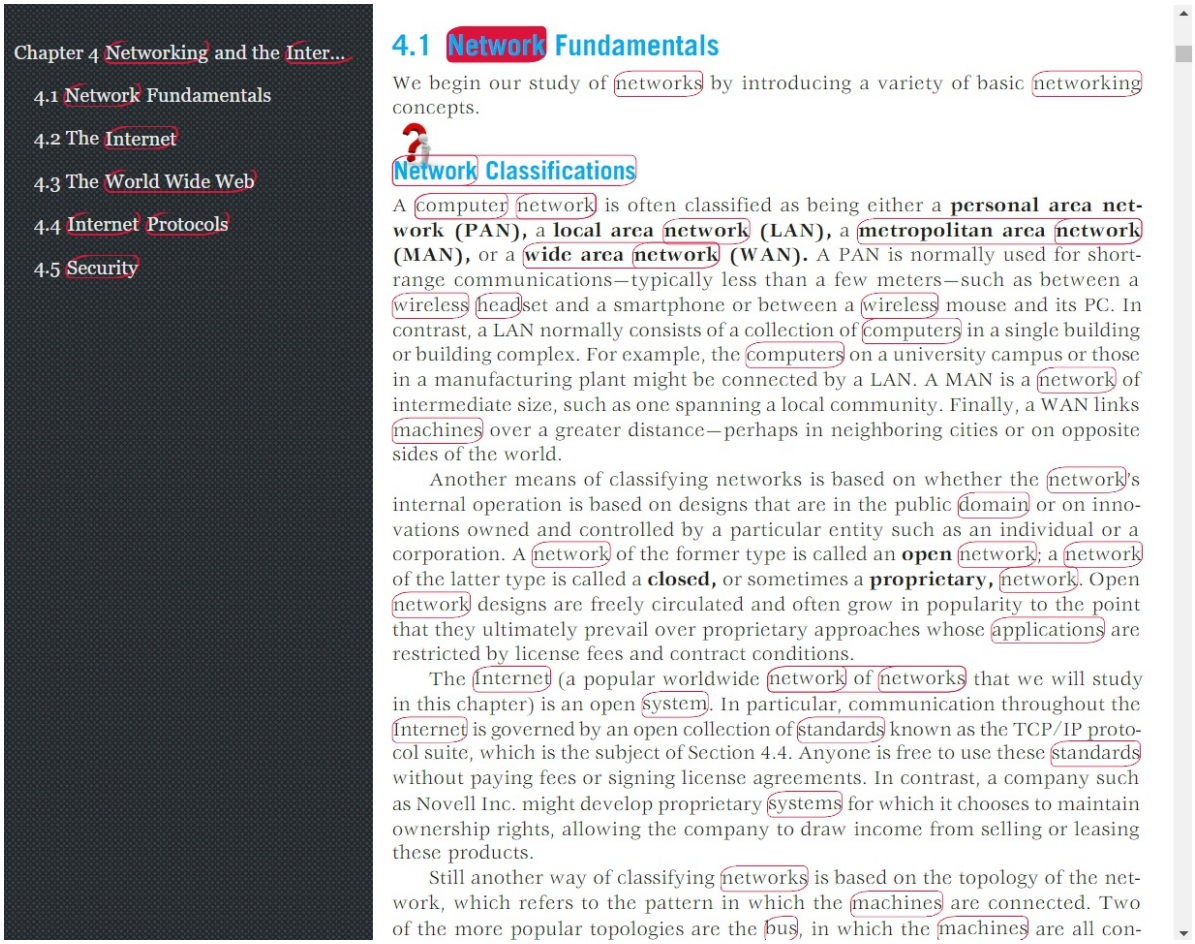

Figure 5.18: Prerequisite enhanced textbook mock-up

CHAPTER 6

CONCLUSIONS, LIMITATIONS AND FUTURE WORKS

Most of the discussions and experiments presented in the previous chapters focused on PR related tasks, as its contextualisation in the field of ILS (intelligent learning systems, see chapter 1), its grounding in pedagogical theories (section 2.1), its formal definition in Knowledge Representation (section 2.2.1), its manual annotation in textbooks (section 5.1), its analysis (section 5.2), its automatic extraction (sections 2.3 and 5.4) and visualisation (section 5.5). Putting it all together, these issues represent one of the most basic and yet critical steps in building ILS, and they are often required in conjunction with other components in the development process. Despite the founding nature of these tasks, researchers' efforts in PR related tasks may unfortunately fail to be appreciated by a final user. For a student or a teacher, other educational technologies or other components in the architecture of an ILS might very easily look more appealing (e.g. an intelligent user interface, an augmented reality enhanced environment, a learning dashboard, etc.). Nevertheless, for the reasons that we already explained in chapter 1, the achievement of a solid knowledge model (with knowledge components and prerequisite relations) constitutes the foundation of a well-designed ILS. At the moment we are working on a possible practical application of the topics covered in this thesis, within the area of Intelligent Textbooks which gained renewed interest in the last years (see chapter 1). More specifically, this is for now a mock-up for a prerequisite-enhanced textbook (a screenshot is show in fig. 5.18), which incorporates prerequisite concept maps and other rich visualisations so that the knowledge structure of the textbook can be examined by teachers and students and they can receive contextual help concerning pedagogical

dependencies of a clicked concept in the linguistic context where it appears.

If we look at the framework as a whole and at its single modules, we can draw contributions and limits for each of them.

The Framework. As a support for the community of researchers working on the analysis and identification of prerequisite relations, we proposed a framework (see chapter 4). We conceived this framework to keep it as coherent as possible from the point of view of a possible user, that may want to upload a text, manually annotate it or extract relations from it, merge different annotations and compute agreement, analyse and visualise annotated datasets, use them to train algorithms, all in a logical order. However, due to practical issues, its development was not linear in all its parts. The idea of the framework comes indeed from the original research problem of our research group. This was the automatic extraction of PR from textbooks, but soon the problem became larger as we realised that for achieving our goal we first needed to solve in advance many other issues, which in turn proved to be not easy to address. For example, an annotated dataset was required, and this temporarily shifted our attention towards the problem of the annotation. Here, the scarce availability of experts, as well as the time-consuming nature of PR manual annotation and the inter annotator agreement issue, constitute a particularly relevant bottleneck throughout all the process, obviously affecting the readiness in the creation of a gold dataset. This also explains why the extraction experiment described in 5.4 was not conducted using a gold standard but asking experts to blindly validate a reasonable sample of the extracted relations. This issue is currently undertaken by our group, in order to allow a further validation of our burst-based method described in section 5.4.

As discussed in Chapter 3, the main contribution of the framework is to provide a rich set of tools, a methodology and guidelines to researchers in ILS that need to manage prerequisite relations in textbooks. This is the distinctive feature of the framework, but at the same time it represents also the limit of its applicability to wider annotation tasks that are not anchored to the text. However, in this respect, it is worth noting that while the annotation phase is anchored to the text, offering a rich resource for linguistic analysis, the concept pairs can always be cleaned from the linguistic context, if needed, and returned as binary relations. Currently the framework has not been tested with an experimental evaluation about its usability and its ability to satisfy the requirements of annotators on specific annotation goals and in different domains. This could show further needs and functionalities that are not available yet.

Annotation, analysis and combination modules. Corpora studies exploit language resources that are created by enriching texts with linguistic information (see e.g. treebanks). Dataset used for automatic prerequisite learning have rarely used this approach based on an explicit linguistic annotation of the text. As a contribution of our work, we presented a methodology to produce datasets following a PR annotation protocol that covers annotation, analysis, revision and combination of individual annotations. Outcomes of our work include annotation guidelines and a PR annotation code book. We believe that applying the approach of corpora studies to this task can highly benefit prerequisite knowledge engineering.

An open issue in our annotation methodology is the management of non-prerequisites. Currently we assume as non-prerequisite relations those that are not explicitly annotated as relations. However, this might be misleading, since the lack of a positive relation between two concepts in an annotated dataset can have two meanings: either (*i*) the annotator looked at those two concepts and decided that a prerequisite relation did not exist, or (*ii*) the annotator did not look at them so PR may or may not exist. A portion of the relations in (*ii*) can be partially solved by leveraging the anti-symmetric property (see 2.2.1): if an expert inserted $A \prec B$, then he probably decided that $B \prec A$ did not exist. On the other hand, a harder case is represented by non-prerequisites that are transitive edges with respect to other relations inserted in the concept graph. In other words, if an expert inserted $A \prec B$ and $B \prec C$ but not $A \prec C$, the latter is a transitive relation that do hold from a conceptual point of view and also according to the PR formal properties, but for some reason the expert decided not to explicitly include it. We showed in 5.5 that, by analysing subgraphs belonging to different experts and investigating their PR annotation behaviour, we can get an intuition of the motivations behind their choices. In particular, their decisions regarding transitive relations may also rise from a different interpretation of what a distant or weak prerequisite relation is. We reported that some of the experts tend to think in terms of graph paths, while others in terms of didactic sequences, and this affects the connectivity of the graph and thus the number of transitive closures. Besides annotation, transitive relations can raise issues concerning combination and extraction as well. For example, combination could be applied to individual graphs with a different number of transitives in order to obtain a more consistent dataset in terms of transitivity, but this could be risky for the issues above. Despite that, the approach could face another issue that concerns manual annotation: even if strict guidelines are provided regarding transitives, annotators may not easily control their behaviour addressing such relations. They may not, in other words, be able to consistently insert all or none of the

transitives, but they will instead insert a differing share of such relations, depending on the linguistic context of the annotation and also on their subjectivity. In the future it could be interesting to investigate the impact of linguistic phenomena on the annotation of transitives. In the present work (see 5.2) we performed graph analysis and studied how gold standards vary when we include a different number of transitive relations. We defined transitives edges by taking into account the concept map diameter, while in the future we plan to test more graph metrics to handle this issue.

Extraction module. The extraction module present our main scientific contribution to the literature, which includes the burst-based method for PR identification (see 5.4.1), its two experimental evaluations (5.4.2 and 5.4.3.1) and the explorative method that uses a neural classifier (briefly presented in 5.4.3.2). The strong point of both methods is that, contrary to most approaches for prerequisite extraction, they are able to extract PR from unstructured text, without exploiting external structured knowledge. The first is an unsupervised method based on burst analysis, co-occurrence, and temporal reasoning for PR relation extracted from educational materials. Current results in terms of precision and recall are encouraging, however, as discussed above, further comparative evaluations are needed to draw conclusions. Regarding this method, our future work includes also refining the PR formula, applying the method to different domains and types of resources, and evaluating it on larger corpora. Moreover, we are confident that a major improvement of the method can be obtained by taking into account more complex patterns and not only making one-by-one comparisons of pairs of intervals. By taking benefit from this knowledge we could enhance the method with a much deeper analysis of how the relation between two concepts evolves across time. Further research directions include analysing and interpreting annotators agreement and improving the burst-based algorithm performance (in particular its precision) with machine learning methods: starting with the annotation described in 5.1.1, we are collecting annotated data for building a gold dataset of educational resources annotated with PR relations that can be used to tune the weights of the temporal patterns. Finally, concerning the evaluation methodology (see 5.4.2), a future line of research is refining the rewarding criterion that we used to evaluate PR paths. As described in section 5.4.2, for evaluating the extracted graph we took into account PRs paths. We did so in order not to penalise the extraction of relations that are implied in the gold standard but not explicitly coded by annotators. This choice is consistent with the formal properties of PR (where transitivity holds by definition) and is supported by similar choices taken in the field of temporal information processing, where evaluation of extracted relations poses similar issues and has been

addressed in similar way. Nevertheless, considering transitivity and path similarity constitutes a first step towards the definition of a more sophisticated rewarding strategy, which represents our next goal. In particular, we recognise that transitive relations might not be always equally informative from a pedagogical point of view. For instance, a transitive relation such as [*information* ≺...≺ *dotted decimal notation* ≺...≺ *IP address*] links two edges (*information* and *IP address*) which however are not deeply pedagogically related. However, making finer distinctions in this sense is not trivial, and requires deep studies on the holistic nature of PR annotation and on the experts behaviour when they decide whether to introduce transitives and PR paths in the concept graph.

The other method is a neural model for prerequisite relation extraction from educational texts. Results are promising also in this case (details reported in [13]), although further work needs to be done. In addition to the extension proposed in the paragraph *Classifier and burst intervals* of section 5.4.3.2 (i.e. using Allen's temporal algebra also in the workflow of this method), we plan further work particularly for improving the performances of the method in a out-of-domain scenario, namely using concept pairs of a different domain during testing.

Visualisation module. Finally, we can draw contributions and limits of our visualisation techniques in the homonymous module. They have been conceived to help researchers and analysts in their effort of better understanding the issue of prerequisite dependencies in textbooks and developing more powerful strategies for the automatic extraction. To the best of our knowledge, this is the first set of visualization techniques to explore PR relations allowing filters and multiple views. Our own use of the module allowed to support the hypotheses regarding the correlation between PR direction and temporal concept ordering, as discussed in 5.5. Furthermore, visual analysis of the PR algorithm provides valid insights on the burst patterns combination. Finally, we are also working on techniques that more directly address the needs of learners and teachers in their common activities of selecting, accessing, exploring and organising learning materials, with our final aim of giving a contribution to the field of intelligent textbooks.

APPENDIX A

CODE BOOK

In the following pages we report a table with coding choices emerged during different iterations of the PR annotation protocol (see 5.1 and subsequent sections). More specifically, examples listed here are drawn from textbook chapters that deal with computer architectures, programming languages, data representation, software engineering, computer graphics, algorithms.

Description	**Examples and comments**
Absent PRs Annotate PRs only when they are present, not when they do not exist (i.e. negative or absent PRs).	*It is necessary to consider two of the special purpose registers within the CPU: the **instruction register** and the **program counter.*** The fact that between ***instruction register*** *and* ***program counter*** there are no PRs must not be coded.
Background knowledge Do not rely on background knowledge, i.e. only encode knowledge that can be directly found in the text.	*The most prominent programming paradigm in today's software development is the **object-oriented paradigm.** Following this paradigm, a software system is viewed as a collection of units, called objects.* Although one can consider **imperative paradigm** as a prerequisite to learn **object-oriented paradigm**, this relation cannot be derived from the quoted text, hence must not be annotated.
Acronyms Do not introduce PRs between an acronym and its expansion.	*This ability of a controller to access main memory is known as **direct memory access** and it is often referred as **DMA**.* Although knowing the expanded version can be useful to guess the meaning of **DMA**, they both refer to the same concept, so there is not a PR.
Hierarchical relations Hypernymy (or "is-a"), meronym ("is-part-of") are generally good indicators of PRs.	*This technique is called **data compression** and can be divided into two categories: **lossless compression** and **lossy compression**.*
Synonyms Do not add PRs between synonyms (they are often signaled by patterns such as "also known as")	*The **imperative paradigm**, also known as the **procedural paradigm**, represents the traditional approach to the programming process.*
Co-requisites Do not introduce PRs between co-requisites, i.e. entities at the same conceptual level (e.g. sibling nodes in a taxonomy or ontology).	***Declarative statements** define customized terminology that is used later in the program; **imperative statements** describe steps in the underlying algorithms; and **comments** enhance the readability.* The fact that these three concepts are listed in a certain order does not establish particular conceptual hierarchies, so there are no PRs.

Co-requisites part 2 (particular case) Despite what is observed in the previous rule, PRs between co-requisites can be traced when a concept is defined in terms of similarity and/or contrast with its siblings that were previously explained.	*Communication between computing devices is handled over two types of paths:* ***parallel*** *and* ***serial communication****. In the case of* ***parallel communication****, [...]. In contrast,* ***serial communication*** *[...]* In the unquoted excerpts an explanation for **parallel communication** is given, then the writer heavily relies on such explanation to explain **serial communication**.
Multi-term concepts In case of multi-term concepts, the PR usually involves their entire span.	***Rotation delay*** *is a measurement used to evaluate a* ***hard disk****'s performance.* If here there is a PR, it is between **hard disk** (not just **disk**) and **rotation delay**.
Compound terms These are multi-terms derived from a composition of two or more terms where at least one is a concept. The full nominal structure is often involved in a PR with one of its elements (e.g. the concept represented by the head or the modifier of the compound).	*1) The leftmost* ***bit*** *is called the* ***most significant bit.*** *2)* ***Registers*** *are data storage cells similar to main memory cells that are used for temporary storage of information within the CPU. The* ***register unit*** *contains such cells and represent one of the element of the CPU.* In the first example the prerequisite is represented by the head (**bit**) of the compound, while in the second example by its modifier (**register**).
Definitions Definition are in general good candidate contexts where we can find PRs, since they usually rely on previously explained concepts.	***Integrated development environments*** *(IDEs) are systems that are used for software development and combine tools such as editors,* ***compilers*** *and* ***debuggers*** *into a single, integrated package.*
Syntactic patterns Patterns (e.g. "called", "such as") may often reveal a PR, although we can easily find also counter-examples, either because one of the nominal entities tied by the pattern is not a concept at all, or because the pattern rather indicates the absence of a PR (e.g. "also known as").	*1) This type of* ***loop*** *is often called a* ***for-each loop*** *in languages other than Python.* *2) Mathematicians refer to such entities as functions, which is the reason this approach is called the* ***functional paradigm****.* The first sentence offers a good example of the use of "(is) called" pattern for expressing PR, while the second represents a counter-example.

Causal relations When there is a direct cause-effect relation between two concepts, a PR can be traced.	*This impediment is known as the **von Neumann bottleneck** because it is a consequence of the underlying **von Neumann architecture** in which a CPU fetches its instructions.*
Generic terms vs primary notions of the domain Do not confuse generic terms with primary notion: the former play no role in the given domain, while the latter are basic concepts of the domain, whose understanding is taken for granted by the writer.	*1) One means of representing an image is to interpret the image as a collection of dots, called **pixels**.* *2) It is because of this repeated insertion process that the underlying algorithm is called the **insertion sort**.* ***Dot*** is not a prerequisite of ***pixel***, since it is not even a specific domain concept but only a generic term that is used in the definition of **pixel**. On the contrary, in the second example, the term algorithm, although somehow general compared to other terms, plays the role of a primary notion in the domain.
Titles Generally a simple mention of a concept in the title is not enough to establish PRs.	**Multiprocessor Machines** Pipelining can be viewed as a first step toward **parallel processing**. Although the title ***Multiprocessor Machines*** prepares the reader to what is about to be explained, there is not a PR with ***parallel processing***, because it merely introduces the topic of the paragraph.
Inter-domain PRs Concepts from other domains can appear as prerequisites in related topics (e.g. mathematics concepts in a computer science textbook).	*All algorithms whose graphs have the shape of a **parabola**, such as the insertion sort, have **quadratic time complexity**.* *To decode a byte expressed in **floating-point notation**, we first extract the **mantissa**.*

APPENDIX B

ANNOTATION GUIDELINES

B.1 Guidelines and suggestions for annotators

Please read carefully the following guidelines before doing the annotation.

1. The goal of the annotation is identifying a prerequisite relation between two distinct terms in a textual corpus. These two terms represent concepts described in the text and can be referred as target concept and prerequisite concept.
2. A concept can be either a single or multi-word term extracted from the corpus.
3. Insert a prerequisite relation for a target concept if you think that you need to know the information related to a different concept in order to understand what you are reading about the target concept. Each of the two concepts must be present either in the initial terminology or in the manual terminology that you built during the annotation process. If a concept is still missing in the terminology, add the corresponding term and then insert the relation.
4. The relation must be inserted exactly in the context (i.e. the sentence) where you find it: if you think that the mention of a target concept recalls information related to another concept, enrich that mention by building the prerequisite pair.
5. Build a concept pair only if a prerequisite relation does exist between the two: if you think that a relation between two concepts is never found in the text, do not insert any relation.

6. Trust the text: you must annotate only concepts and relations that you can find in the text. Do not consider concepts and relations that you may only recall from your background knowledge about the topic.

7. A concept cannot be prerequisite of itself: self prerequisites such as "*computer* is a prerequisite of *computer*" will not be allowed by the system.

8. Do not introduce loops in the annotation. Imagine that you have already annotated that: 1) "fruit" is a prerequisite of "citrus", and 2) "citrus" is a prerequisite of "orange". By annotating that "orange" is a prerequisite of "fruit", you will create a loop.

9. Every time you insert a relation you must also define its weight. Allowed values comprise: *strong* (the prerequisite is absolutely necessary to understand the other term) and *weak* (the prerequisite is very useful but not strictly necessary).

10. Delete a prerequisite relation if you added it by mistake. Keep in mind, however, that you can delete one single instance of a pair at a time: if the same pair is annotated with the prerequisite relation in another part of the text, that relation will be preserved. If you think that ALL prerequisite relations between two given concepts should be deleted, you must delete each of the relations having those two concepts.

B.2 Knowledge Elicitation Questions

If you experience difficulty when trying to understand what a prerequisite concept can be for a given target concept, try to ask yourself the following questions:

1. Which concepts (among those mentioned in the text) you need to master in order to understand the meaning of the target concept?

2. Which concepts are recalled in the text to define the target concept?

3. Are other concepts mentioned in the same context (e.g. sentence or paragraph) of the target concept? If so, are they necessary to understand the meaning of the target concept?

4. Try to follow the expository flow provided by the author(s) of the text. According to that, does the target concept represent a special case of another concept mentioned in the text? (e.g. *circumference*[target] is a special case of *ellipsis*[prerequisite].

5. Does the target concept denote a part of a bigger element which is denoted by another concept mentioned in the text? (e.g. the *elbow*[target] is a part of an *arm*[prerequisite]).

6. Does the target concept consist of sub-elements that are mentioned in the text? (e.g. *elbow*, *forearm* and *shoulder*[prerequisites] are parts of the *arm*[target]).

7. Is the target concept caused by another previously described concept (or viceversa)? (e.g. *rain*[prerequisite] causes *floodings*[target], or *rain*[target] is caused by *low pressure*[prerequisite]). Again, as in the previous cases, try to follow the relation proposed by the text author.

BIBLIOGRAPHY

[1] S. A. ADJEI, A. F. BOTELHO, AND N. T. HEFFERNAN, *Predicting student performance on post-requisite skills using prerequisite skill data: an alternative method for refining prerequisite skill structures*, in Proceedings of the sixth international conference on learning analytics & knowledge, 2016, pp. 469–473.

[2] G. ADORNI, C. ALZETTA, F. KOCEVA, S. PASSALACQUA, AND I. TORRE, *Towards the identification of propaedeutic relations in textbooks*, in International Conference on Artificial Intelligence in Education, Springer, 2019, pp. 1–13.

[3] G. ADORNI, F. DELL'ORLETTA, F. KOCEVA, I. TORRE, AND G. VENTURI, *Extracting dependency relations from digital learning content*, in Italian Research Conference on Digital Libraries, Springer, 2018, pp. 114–119.

[4] G. ADORNI AND F. KOCEVA, *Designing a knowledge representation tool for subject matter structuring*, in International Workshop on Graph Structures for Knowledge Representation and Reasoning, Springer, 2015, pp. 1–14.

[5] ——, *Educational concept maps for personalized learning path generation*, in Conference of the Italian Association for Artificial Intelligence, Springer, 2016, pp. 135–148.

[6] R. AGRAWAL, B. GOLSHAN, AND E. PAPALEXAKIS, *Toward data-driven design of educational courses: a feasibility study*, Journal of Educational Data Mining, 8 (2016), pp. 1–21.

[7] V. ALEVEN, J. SEWALL, O. POPESCU, F. XHAKAJ, D. CHAND, R. BAKER, Y. WANG, G. SIEMENS, C. ROSÉ, AND D. GASEVIC, *The beginning of a beautiful friendship? intelligent tutoring systems and MOOCs*, in International Conference on Artificial Intelligence in Education, Springer, 2015, pp. 525–528.

[8] J. F. ALLEN, *Maintaining knowledge about temporal intervals*, Communications of the ACM, 26 (1983).

[9] I. ALPIZAR-CHACON, Ö. ERENSOY, AND S. SOSNOVSKY, *Order out of chaos: Construction of knowledge models from pdf textbooks*, in Proceedings of the 10th International Conference on Knowledge Capture (Submitted)(K-CAP'19). ACM, New York, NY, USA, 2019.

[10] I. ALPIZAR-CHACON AND S. SOSNOVSKY, *Interlingua: Linking textbooks across different languages*, in First Workshop on Intelligent Textbooks, 2019.

[11] F. ALSAAD, A. BOUGHOULA, C. GEIGLE, H. SUNDARAM, AND C. ZHAI, *Mining MOOC lecture transcripts to construct concept dependency graphs*, in Proceedings of the 11th International Conference on Educational Data Mining, EDM 2018, Buffalo, NY, USA, July 15-18, 2018.

[12] C. ALZETTA, F. KOCEVA, S. PASSALACQUA, I. TORRE, AND G. ADORNI, *PRET: Prerequisite-enriched terminology. a case study on educational texts*, in Proceedings of the Fifth Italian Conference on Computational Linguistics, 2018.

[13] C. ALZETTA, A. MIASCHI, G. ADORNI, F. DELL'ORLETTA, F. KOCEVA, S. PASSALACQUA, AND I. TORRE, *Prerequisite or not prerequisite? That's the problem! An NLP-based approach for concept prerequisites learning*, in Proceedings of the Sixth Italian Conference on Computational Linguistics, 2019.

[14] L. AROYO, P. DOLOG, G.-J. HOUBEN, M. KRAVCIK, A. NAEVE, M. NILSSON, AND F. WILD, *Interoperability in personalized adaptive learning*, Journal of Educational Technology & Society, 9 (2006), pp. 4–18.

[15] R. ARTSTEIN AND M. POESIO, *Inter-coder agreement for computational linguistics*, Computational Linguistics, 34 (2008), pp. 555–596.

[16] N. ASGHAR, *Automatic extraction of causal relations from natural language texts: a comprehensive survey*, arXiv preprint arXiv:1605.07895, (2016).

[17] M. N. ASIM, M. WASIM, M. U. G. KHAN, W. MAHMOOD, AND H. M. ABBASI, *A survey of ontology learning techniques and applications*, Database, 2018 (2018).

[18] I. AUGENSTEIN, M. DAS, S. RIEDEL, L. VIKRAMAN, AND A. MCCALLUM, *SemEval 2017 task 10: ScienceIE - Extracting keyphrases and relations from scientific publications*, in Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), 2017, pp. 546–555.

[19] D. P. AUSUBEL, *The psychology of meaningful verbal learning*, (1963).

[20] ——, *In defense of advance organizers: A reply to the critics*, Review of Educational research, 48 (1978), pp. 251–257.

[21] D. P. AUSUBEL, J. D. NOVAK, H. HANESIAN, ET AL., *Educational psychology: A cognitive view*, (1968).

[22] R. S. BAKER, *Stupid tutoring systems, intelligent humans*, International Journal of Artificial Intelligence in Education, 26 (2016), pp. 600–614.

[23] T. BARNES, *The Q-matrix method: Mining student response data for knowledge*, in American Association for Artificial Intelligence 2005 Educational Data Mining Workshop, 2005, pp. 1–8.

[24] L. L. BELGRAVE AND K. J. SMITH, *Negotiated validity in collaborative ethnography*, Qualitative Inquiry, 1 (1995), pp. 69–86.

[25] E. M. BENNETT, R. ALPERT, AND A. GOLDSTEIN, *Communications through limited-response questioning*, Public Opinion Quarterly, 18 (1954), pp. 303–308.

[26] J. BINDER, *Package 'bursts': Markov model for bursty behavior in streams*, 2014. R package version 1.0-1.

[27] E. BLANCO, N. CASTELL, AND D. I. MOLDOVAN, *Causal relation extraction*, in LREC, 2008.

[28] D. M. BLEI, A. Y. NG, AND M. I. JORDAN, *Latent Dirichlet Allocation*, Journal of machine Learning research, 3 (2003), pp. 993–1022.

[29] B. S. BLOOM, *Learning for mastery. instruction and curriculum. regional education laboratory for the carolinas and virginia, topical papers and reprints, number 1*, Evaluation comment, 1 (1968), p. n2.

[30] ——, *Mastery learning*, Mastery learning: Theory and practice, (1971), pp. 47–63.

[31] B. S. BLOOM, M. ENGLEHART, E. J. FURST, W. H. HILL, AND D. R. KRATHWOHL, *Taxonomy of educational objectives: Handbook i*, Cognitive domain. New York: David McKay, (1956).

[32] F. BONIN, F. DELL'ORLETTA, G. VENTURI, AND S. MONTEMAGNI, *A contrastive approach to multi-word term extraction from domain corpora*, in Proceedings of the 7th International Conference on Language Resources and Evaluation, 2010.

[33] J. L. BORGES, *La biblioteca de babel*, Obras completas, 1 (1941).

[34] D. BOULANGER AND V. KUMAR, *An overview of recent developments in intelligent e-Textbooks and reading analytics*, in First Workshop on Intelligent Textbooks, 2019.

[35] G. E. BOX, *Science and statistics*, Journal of the American Statistical Association, 71 (1976), pp. 791–799.

[36] R. J. BRACHMAN, H. J. LEVESQUE, AND R. REITER, *Knowledge representation*, MIT press, 1992.

[37] G. BROOKSHEAR AND D. BRYLOW, *Computer Science: An Overview, Global Edition*, Pearson Education Limited., 2015, ch. 4 Networking and the Internet.

[38] J. S. BRUNER, *The act of discovery*, (1961).

[39] P. BRUSILOVSKY, *Developing adaptive educational hypermedia systems: From design models to authoring tools*, in Authoring tools for advanced technology Learning Environments, Springer, 2003, pp. 377–409.

[40] P. BRUSILOVSKY AND E. MILLÁN, *User models for adaptive hypermedia and adaptive educational systems*, in The adaptive web, Springer, 2007, pp. 3–53.

[41] P. BRUSILOVSKY AND J. VASSILEVA, *Course sequencing techniques for large-scale web-based education*, Int. Journal of Continuing Engineering Education and Life-long Learning, (2002).

[42] P. BRUSILOVSKY, M. YUDELSON, AND S. SOSNOVSKY, *An adaptive e-learning service for accessing interactive examples*, in E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, Association for the Advancement of Computing in Education (AACE), 2004, pp. 2556–2561.

[43] S. BUCHHOLZ AND E. MARSI, *CoNLL-X shared task on multilingual dependency parsing*, in Proceedings of the tenth conference on computational natural language learning (CoNLL-X), 2006, pp. 149–164.

[44] P. BUITELAAR, P. CIMIANO, AND B. MAGNINI, *Ontology learning from text: An overview*, Ontology learning from text: Methods, evaluation and applications, 123 (2005), pp. 3–12.

[45] J. CAMACHO-COLLADOS, C. D. BOVI, L. E. ANKE, S. ORAMAS, T. PASINI, E. SANTUS, V. SHWARTZ, R. NAVIGLI, AND H. SAGGION, *SemEval-2018 task 9: Hypernym discovery*, in Proceedings of The 12th International Workshop on Semantic Evaluation, 2018, pp. 712–724.

[46] S. CAREY, *The origin of concepts*, Oxford University Press, 2009.

[47] C. CARMONA, E. MILLÁN, J.-L. PÉREZ-DE-LA CRUZ, M. TRELLA, AND R. CONEJO, *Introducing prerequisite relations in a multi-layered Bayesian student model*, in International Conference on User Modeling, Springer, 2005, pp. 347–356.

[48] J. B. CARROLL, *A model of school learning*, Teachers college record, (1963).

[49] P. F. CARVALHO, M. GAO, B. A. MOTZ, AND K. R. KOEDINGER, *Analyzing the relative learning benefits of completing required activities and optional readings in online courses*, International Educational Data Mining Society, (2018).

[50] M. T. C. CASTELLVÍ, *Terminology: Theory, methods and applications*, vol. 1, John Benjamins Publishing, 1999.

[51] J. M. CEJUELA, P. MCQUILTON, L. PONTING, S. J. MARYGOLD, R. STEFANCSIK, G. H. MILLBURN, AND B. ROST, *tagtog: interactive and text-mining-assisted annotation of gene mentions in PLOS full-text articles*, Database, 2014 (2014).

[52] S. CHANGUEL AND N. LABROCHE, *Distinguishing defined concepts from prerequisite concepts in learning resources*, in 2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), IEEE, 2011, pp. 22–29.

[53] S. CHANGUEL, N. LABROCHE, AND B. BOUCHON-MEUNIER, *Resources sequencing using automatic prerequisite–outcome annotation*, ACM Transactions on Intelligent Systems and Technology (TIST), 6 (2015), p. 6.

[54] D. S. CHAPLOT, Y. YANG, J. G. CARBONELL, AND K. R. KOEDINGER, *Data-driven automated induction of prerequisite structure graphs*, in EDM, 2016, pp. 318–323.

[55] N.-S. CHEN, C.-W. WEI, H.-J. CHEN, ET AL., *Mining e-learning domain concept map from academic articles*, Computers & Education, 50 (2008), pp. 1009–1021.

[56] Y. CHEN, P.-H. WUILLEMIN, AND J.-M. LABAT, *Bayesian student modeling improved by diagnostic items*, in International Conference on Intelligent Tutoring Systems, Springer, 2014, pp. 144–149.

[57] Y.-L. CHI, *Ontology-based curriculum content sequencing system with semantic rules*, Expert Systems with Applications, 36 (2009), pp. 7838–7847.

[58] P. CIMIANO, A. HOTHO, AND S. STAAB, *Learning concept hierarchies from text corpora using formal concept analysis*, Journal of artificial intelligence research, 24 (2005), pp. 305–339.

[59] P. CIMIANO AND J. VÖLKER, *text2onto*, in International conference on application of natural language to information systems, Springer, 2005, pp. 227–238.

[60] R. B. CLARIANA AND R. KOUL, *A computer-based approach for translating text into concept map-like representations*, in Proceedings of the first international conference on concept mapping, 2004, pp. 14–17.

[61] C. COFFRIN, L. CORRIN, P. DE BARBA, AND G. KENNEDY, *Visualizing patterns of student engagement and performance in MOOCs*, in Proceedings of the fourth international conference on learning analytics and knowledge, ACM, 2014, pp. 83–92.

[62] J. COHEN, *A coefficient of agreement for nominal scales*, Educational and psychological measurement, 20 (1960), pp. 37–46.

[63] M. COLE, *Using wiki technology to support student engagement: Lessons from the trenches*, Computers & education, 52 (2009), pp. 141–146.

[64] A. T. CORBETT AND J. R. ANDERSON, *Knowledge tracing: Modeling the acquisition of procedural knowledge*, User modeling and user-adapted interaction, 4 (1994), pp. 253–278.

[65] J. CORTÁZAR, J. ORTEGA, AND S. YURKIÉVICH, *Rayuela*, vol. 16, EdUSP, 1996.

[66] R. S. D BAKER, A. T. CORBETT, AND V. ALEVEN, *More accurate student modeling through contextual estimation of slip and guess probabilities in Bayesian knowledge tracing*, in International conference on intelligent tutoring systems, Springer, 2008, pp. 406–415.

[67] S. C. DE ABREU, T. L. BONAMIGO, AND R. VIEIRA, *A review on relation extraction with an eye on portuguese*, Journal of the Brazilian Computer Society, 19 (2013), p. 553.

[68] C. DE MEDIO, F. GASPARETTI, C. LIMONGELLI, F. SCIARRONE, AND M. TEMPERINI, *A machine learning approach to identify dependencies among learning objects*, in CSEDU (1), 2016, pp. 345–352.

[69] F. DELL'ORLETTA, G. VENTURI, A. CIMINO, AND S. MONTEMAGNI, *T2k^2: a system for automatically extracting and organizing knowledge from texts*, in Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), 2014.

[70] V. DEVEDZIC, *Education and the semantic web*, International Journal of Artificial Intelligence in Education, 14 (2004), pp. 165–191.

[71] P. DILLENBOURG, *Orchestration graphs*, no. BOOK, EPFL press, 2015.

[72] J.-P. DOIGNON AND J.-C. FALMAGNE, *Spaces for the assessment of knowledge*, International journal of man-machine studies, 23 (1985), pp. 175–196.

[73] M. P. DRISCOLL, *Psychology of learning for instruction*, Allyn & Bacon, 1994.

[74] H. DUFORT, E. AÏMEUR, C. FRASSON, AND M. LALONDE, *Curriculum evaluation: A case study*, in Intelligent Tutoring Systems: 4th International Conference, ITS'98, San Antonio, Texas, USA, August 16–19, 1998, Proceedings, Springer, 2003, p. 106.

[75] E. DUVAL, *Attention please! Learning analytics for visualization and recommendation*, LAK, 11 (2011), pp. 9–17.

[76] R. ECKART DE CASTILHO, É. MÚJDRICZA-MAYDT, S. M. YIMAM, S. HARTMANN, I. GUREVYCH, A. FRANK, AND C. BIEMANN, *A web-based tool for the integrated annotation of semantic and syntactic structures*, in Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH), Osaka, Japan, Dec. 2016, The COLING 2016 Organizing Committee, pp. 76–84.

[77] B. D. EUGENIO AND M. GLASS, *The kappa statistic: A second look*, Computational linguistics, 30 (2004), pp. 95–101.

[78] A. R. FABBRI, I. LI, P. TRAIRATVORAKUL, Y. HE, W. T. TING, R. TUNG, C. WESTERFIELD, AND D. R. RADEV, *TutorialBank: A manually-collected corpus for prerequisite chains, survey extraction and resource recommendation*, in ACL, 2018.

[79] J.-C. FALMAGNE, D. ALBERT, C. DOBLE, D. EPPSTEIN, AND X. HU, *Knowledge spaces: Applications in education*, Springer Science & Business Media, 2013.

[80] J.-C. FALMAGNE AND J.-P. DOIGNON, *Learning spaces: Interdisciplinary applied mathematics*, Springer Science & Business Media, 2010.

[81] D. FAURE AND C. NEDELLEC, *Knowledge acquisition of predicate argument structures from technical texts using machine learning: The system asium*, in International Conference on Knowledge Engineering and Knowledge Management, Springer, 1999, pp. 329–334.

[82] C. J. FILLMORE ET AL., *Frame semantics*, Cognitive linguistics: Basic readings, 34 (2006), pp. 373–400.

[83] M. A. FINLAYSON AND T. ERJAVEC, *Overview of annotation creation: Processes and tools*, in Handbook of Linguistic Annotation, Springer, 2017, pp. 167–191.

[84] J. L. FLEISS, *Measuring nominal scale agreement among many raters*, Psychological bulletin, 76 (1971), p. 378.

[85] K. FORT, A. NAZARENKO, AND S. ROSSET, *Modeling the complexity of manual annotation tasks: a grid of analysis*, in International Conference on Computational Linguistics, 2012, pp. 895–910.

[86] K. FRANTZI AND S. ANANIADOU, *The C-value/NC-value domain independent method for multi-word term extraction*, Journal of NLP, 6 (1999), pp. 145–179.

[87] K. FRANTZI, S. ANANIADOU, AND H. MIMA, *Automatic recognition of multi-word terms:. the C-value/NC-value method*, International journal on digital libraries, 3 (2000), pp. 115–130.

[88] R. FU, J. GUO, B. QIN, W. CHE, H. WANG, AND T. LIU, *Learning semantic hierarchies via word embeddings*, in Proceedings of the 52nd Annual Meeting

of the Association for Computational Linguistics (Volume 1: Long Papers), 2014, pp. 1199–1209.

[89] K. GÁBOR, D. BUSCALDI, A.-K. SCHUMANN, B. QASEMIZADEH, H. ZARGAYOUNA, AND T. CHARNOIS, *Semeval-2018 Task 7: Semantic relation extraction and classification in scientific papers*, in Proceedings of International Workshop on Semantic Evaluation (SemEval-2018), New Orleans, LA, USA, 2018.

[90] R. M. GAGNÉ, *The acquisition of knowledge*, Psychological review, 69 (1962), p. 355.

[91] R. M. GAGNÉ, *The conditions of learning*, Holt, Rinehart and Winston, 1965.

[92] R. M. GAGNÉ, *Learning hierarchies*, Educational psychologist, 6 (1968), pp. 1–9.

[93] R. M. GAGNÉ AND R. GLASER, *Foundations in learning research*, Instructional technology: foundations, (1987), pp. 49–83.

[94] F. GASPARETTI, C. DE MEDIO, C. LIMONGELLI, F. SCIARRONE, AND M. TEMPERINI, *Prerequisites between learning objects: Automatic extraction based on a machine learning approach*, Telematics and Informatics, 35 (2018), pp. 595–610.

[95] F. GASPARETTI, C. LIMONGELLI, AND F. SCIARRONE, *Exploiting wikipedia for discovering prerequisite relationships among learning objects*, in 2015 International Conference on Information Technology Based Higher Education and Training (ITHET), IEEE, 2015, pp. 1–6.

[96] A. GLIOZZO AND C. STRAPPARAVA, *Semantic domains in computational linguistics*, Springer Science & Business Media, 2009.

[97] W. GOLIK, R. BOSSY, Z. RATKOVIC, AND C. NÉDELLEC, *Improving term extraction with linguistic analysis in the biomedical domain*, Research in Computing Science, 70 (2013), pp. 157–172.

[98] J. GORDON, S. AGUILAR, E. SHENG, AND G. BURNS, *Structured generation of technical reading lists*, in Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications, 2017, pp. 261–270.

[99] J. GORDON, L. ZHU, A. GALSTYAN, P. NATARAJAN, AND G. BURNS, *Modeling concept dependencies in a scientific corpus*, in Proceedings of the 54th Annual

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, 2016, pp. 866–875.

[100] T. R. GRUBER, *A translation approach to portable ontology specifications*, Knowledge acquisition, 5 (1993), pp. 199–220.

[101] ——, *Toward principles for the design of ontologies used for knowledge sharing?*, International journal of human-computer studies, 43 (1995), pp. 907–928.

[102] M. GUEFFAZ, J. DESLIS, AND J.-C. MOISSINAC, *Curriculum data enrichment with ontologies*, in Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14), 2014, pp. 1–6.

[103] T. HAK AND T. BERNTS, *Coder training: Theoretical training or practical socialization?*, Qualitative Sociology, 19 (1996), pp. 235–257.

[104] S. HÅKLEV, L. FAUCON, T. HADZILACOS, AND P. DILLENBOURG, *Orchestration graphs: Enabling rich social pedagogical scenarios in MOOCs*, in Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale, 2017, pp. 261–264.

[105] K. A. HALLGREN, *Computing inter-rater reliability for observational data: an overview and tutorial*, Tutorials in quantitative methods for psychology, 8 (2012), p. 23.

[106] B. HARRY, K. M. STURGES, AND J. K. KLINGNER, *Mapping the process: An exemplar of process and challenge in grounded theory analysis*, Educational researcher, 34 (2005), pp. 3–13.

[107] M. HAZMAN, S. R. EL-BELTAGY, AND A. RAFEA, *A survey of ontology learning approaches*, International Journal of Computer Applications, 22 (2011), pp. 36–43.

[108] M. A. HEARST, *Automatic acquisition of hyponyms from large text corpora*, in Proceedings of the 14th conference on Computational linguistics-Volume 2, Association for Computational Linguistics, 1992, pp. 539–545.

[109] ——, *Tilebars: visualization of term distribution information in full text information access*, in Proceedings of the SIGCHI conference on Human factors in computing systems, 1995, pp. 59–66.

[110] N. T. HEFFERNAN AND C. L. HEFFERNAN, *The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching*, International Journal of Artificial Intelligence in Education, 24 (2014), pp. 470–497.

[111] F. HILL AND A. KORHONEN, *Learning abstract concept embeddings from multimodal data: Since you probably can't see what I mean*, in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 255–265.

[112] A. HIPPISLEY, D. CHENG, AND K. AHMAD, *The head-modifier principle and multilingual term extraction*, Natural Language Engineering, 11 (2005), pp. 129–157.

[113] R. HUBSCHER, *What's in a prerequisite [learning environments]*, in Proceedings IEEE International Conference on Advanced Learning Technologies, IEEE, 2001, pp. 365–368.

[114] R. J. L. JOHN, T. S. MCTAVISH, AND R. J. PASSONNEAU, *Semantic graphs for mathematics word problems based on mathematics terminology*, in EDM (Workshops), 2015.

[115] D. A. KEIM, F. MANSMANN, J. SCHNEIDEWIND, AND H. ZIEGLER, *Challenges in visual data analysis*, in Tenth International Conference on Information Visualisation (IV'06), IEEE, 2006, pp. 9–16.

[116] F. S. KELLER, *"good-bye, teacher..." 1*, Journal of applied behavior analysis, 1 (1968), pp. 79–89.

[117] B. W. KERNIGHAM AND D. M. RITCHIE, *The C programming language*, Prentice hall of India, 1973.

[118] P. A. KIRSCHNER, J. SWELLER, AND R. E. CLARK, *Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching*, Educational psychologist, 41 (2006), pp. 75–86.

[119] J. KLEINBERG, *Bursty and hierarchical structure in streams*, Data Mining and Knowledge Discovery, 7 (2003), pp. 373–397.

[120] B. KLUGA, M. S. JASTI, V. NAPLES, AND R. FREEDMAN, *Adding intelligence to a textbook for human anatomy with a causal concept map based its*, in First Workshop on Intelligent Textbooks, 2019.

[121] K. R. KOEDINGER, A. T. CORBETT, AND C. PERFETTI, *The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning*, Cognitive science, 36 (2012), pp. 757–798.

[122] L. KOTLERMAN, I. DAGAN, I. SZPEKTOR, AND M. ZHITOMIRSKY-GEFFET, *Directional distributional similarity for lexical inference*, Natural Language Engineering, 16 (2010), pp. 359–389.

[123] S. D. KRASHEN, *The input hypothesis: Issues and implications*, Addison-Wesley Longman Ltd, 1985.

[124] D. R. KRATHWOHL AND L. W. ANDERSON, *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives*, Longman, 2009.

[125] I. LABUTOV, Y. HUANG, P. BRUSILOVSKY, AND D. HE, *Semi-supervised techniques for mining learning outcomes and prerequisites*, in Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2017, pp. 907–915.

[126] J. R. LANDIS AND G. G. KOCH, *The measurement of observer agreement for categorical data*, Biometrics, 33 (1977), pp. 159–174.

[127] G. P. LANDOW, *HyperText: the convergence of contemporary critical theory and technology (parallax: re-visions of culture and society series)*, Johns Hopkins University Press, 1991.

[128] S. LARSEN, *Information can be transmitted but knowledge must be induced*, PLET: Programmed Learning & Educational Technology, 23 (1986), pp. 331–336.

[129] R. Y. LAU, D. SONG, Y. LI, T. C. CHEUNG, AND J.-X. HAO, *Toward a fuzzy domain ontology extraction method for adaptive e-learning*, IEEE transactions on knowledge and data engineering, 21 (2008), pp. 800–813.

[130] S. LEE, Y. PARK, AND W. C. YOON, *Burst analysis for automatic concept map creation with a single document*, Expert Systems with Applications, 42 (2015), pp. 8817–8829.

[131] A. LENCI, *Distributional semantics in linguistic and cognitive research*, Italian journal of linguistics, 20 (2008), pp. 1–31.

[132] H. J. LEVESQUE, *Knowledge representation and reasoning*, Annual review of computer science, 1 (1986), pp. 255–287.

[133] I. LI, A. R. FABBRI, R. R. TUNG, AND D. R. RADEV, *What should i learn first: Introducing LectureBank for NLP education and prerequisite chain learning*, Proceedings of AAAI 2019, (2019).

[134] Y. LI, Z. SHAO, X. WANG, X. ZHAO, AND Y. GUO, *A concept map-based learning paths automatic generation algorithm for adaptive learning systems*, IEEE Access, 7 (2019), pp. 245–255.

[135] C. LIANG, Z. WU, W. HUANG, AND C. L. GILES, *Measuring prerequisite relations among concepts*, in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1668–1674.

[136] C. LIANG, J. YE, S. WANG, B. PURSEL, AND C. L. GILES, *Investigating active learning for concept prerequisite learning*, Proc. EAAI, (2018).

[137] C. LIANG, J. YE, Z. WU, B. PURSEL, AND C. L. GILES, *Recovering concept prerequisite relations from university course dependencies*, in AAAI, 2017, pp. 4786–4791.

[138] C. LIANG, J. YE, H. ZHAO, B. PURSEL, AND C. L. GILES, *Active learning of strict partial orders: A case study on concept prerequisite relations*, arXiv preprint arXiv:1801.06481, (2018).

[139] C. LIMONGELLI, F. GASPARETTI, AND F. SCIARRONE, *Wiki course builder: a system for retrieving and sequencing didactic materials from Wikipedia*, in 2015 International Conference on Information Technology Based Higher Education and Training (ITHET), IEEE, 2015, pp. 1–6.

[140] X. LING AND D. S. WELD, *Temporal information extraction*, in AAAI, vol. 10, 2010, pp. 1385–1390.

[141] H. LIU, W. MA, Y. YANG, AND J. CARBONELL, *Learning concept graphs from online educational data*, Journal of Artificial Intelligence Research, 55 (2016), pp. 1059–1090.

[142] H. LLORENS, N. UZZAMAN, AND J. F. ALLEN, *Merging temporal annotations*, in 2012 19th International Symposium on Temporal Representation and Reasoning, IEEE, 2012, pp. 107–113.

[143] W. LU, Y. ZHOU, J. YU, AND C. JIA, *Concept extraction and prerequisite relation learning from educational data*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 9678–9685.

[144] A. R. LURIA, *Cognitive development: Its cultural and social foundations*, Harvard university press, 1976.

[145] R. MANRIQUE, B. PEREIRA, O. MARINO, N. CARDOZO, AND S. WOLFGAND, *Towards the identification of concept prerequisites via knowledge graphs*, in 2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT), vol. 2161, IEEE, 2019, pp. 332–336.

[146] R. MANRIQUE, J. SOSA, O. MARINO, B. P. NUNES, AND N. CARDOZO, *Investigating learning resources precedence relations via concept prerequisite learning*, in 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), IEEE, 2018, pp. 198–205.

[147] N. MATSUDA AND M. SHIMMEI, *Pastel: Evidence-based learning engineering method to create intelligent online textbook at scale*, (2019).

[148] R. E. MAYER, *Learners as information processors: Legacies and limitations of educational psychology's second metaphor*, Educational psychologist, 31 (1996), pp. 151–161.

[149] A. MCAULEY, B. STEWART, G. SIEMENS, AND D. CORMIER, *The MOOC model for digital practice*, (2010).

[150] P. MCNAMEE, R. SNOW, P. SCHONE, AND J. MAYFIELD, *Learning named entity hyponyms for question answering*, in Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II, 2008.

[151] T. S. MCTAVISH, *Facilitating graph interpretation via interactive hierarchical edges*, in EDM (Workshops), 2014.

[152] R. MENG, S. ZHAO, S. HAN, D. HE, P. BRUSILOVSKY, AND Y. CHI, *Deep keyphrase generation*, arXiv preprint arXiv:1704.06879, (2017).

[153] M. D. MERRILL, R. D. TENNYSON, AND L. O. POSEY, *Teaching concepts: An instructional design guide*, Educational Technology, 1992.

[154] A. MIASCHI, C. ALZETTA, F. A. CARDILLO, AND F. DELL'ORLETTA, *Linguistically-driven strategy for concept prerequisites learning on italian*, in Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, 2019, pp. 285–295.

[155] E. MILIOS, Y. ZHANG, B. HE, AND L. DONG, *Automatic term extraction and document similarity in special text corpora*.

[156] B. MILLER AND D. RANUM, *Runestone interactive: tools for creating interactive course materials*, in Proceedings of the first ACM conference on Learning@ scale conference, ACM, 2014, pp. 213–214.

[157] M. L. MURPHY, *Lexical meaning*, Cambridge University Press, 2010.

[158] T. L. NAPS, G. RÖSSLING, V. ALMSTRUM, W. DANN, R. FLEISCHER, C. HUNDHAUSEN, A. KORHONEN, L. MALMI, M. MCNALLY, S. RODGER, ET AL., *Exploring the role of visualization and engagement in computer science education*, in ACM Sigcse Bulletin, ACM, 2002, pp. 131–152.

[159] R. NAVIGLI AND P. VELARDI, *Learning word-class lattices for definition and hypernym extraction*, in Proceedings of the 48th annual meeting of the association for computational linguistics, Association for Computational Linguistics, 2010, pp. 1318–1327.

[160] M. NEVES AND J. ŠEVA, *An extensive review of tools for manual annotation of documents*, Briefings in Bioinformatics, (2019).

[161] L. NGUYEN AND P. DO, *Learner model in adaptive learning*, World Academy of Science, Engineering and Technology, 45 (2008), pp. 395–400.

[162] P. P. C. R. A. S. NICK DELIGIANNIS, DIONYSIS PANAGIOTOPOULOS, *Interactive and personalized activity ebooks for learning to read: The iread case*, in First Workshop on Intelligent Textbooks, 2019.

[163] R. NKAMBOU, R. MIZOGUCHI, AND J. BOURDEAU, *Advances in intelligent tutoring systems*, vol. 308, Springer Science & Business Media, 2010.

[164] J. D. NOVAK, *Concept mapping: A useful tool for science education*, Journal of research in science teaching, 27 (1990), pp. 937–949.

[165] J. D. NOVAK AND A. J. CAÑAS, *The theory underlying concept maps and how to construct and use them*, research report 2006-01 Rev 2008-01, Florida Institute for Human and Machine Cognition, 2006.

[166] J. D. NOVAK AND A. J. CAÑAS, *The theory underlying concept maps and how to construct and use them*, (2008).

[167] J. D. NOVAK AND D. B. GOWIN, *Learning how to learn*, Cambridge University Press, 1984.

[168] D. L. OLSON AND D. DELEN, *Advanced data mining techniques*, Springer Science & Business Media, 2008.

[169] L. PAN, C. LI, J. LI, AND J. TANG, *Prerequisite relation learning for concepts in MOOCs*, in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, 2017, pp. 1447–1456.

[170] ——, *Prerequisite relation learning for concepts in MOOCs*, (2017).

[171] L. PAN, X. WANG, C. LI, J. LI, AND J. TANG, *Course concept extraction in MOOCs via embedding-based graph propagation*, in Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), vol. 1, 2017, pp. 875–884.

[172] K. PARKER AND J. CHAO, *Wiki as a teaching tool*, Interdisciplinary Journal of e-learning and Learning Objects, 3 (2007), pp. 57–72.

[173] G. PASK, *Conversation, cognition and learning: A cybernetic theory and methodology*, Elsevier Publishing Company, 1975.

[174] S. PASSALACQUA, F. KOCEVA, C. ALZETTA, I. TORRE, AND G. ADORNI, *Visualisation analysis for exploring prerequisite relations in textbooks*, in First Workshop on Intelligent Textbooks, 2019.

[175] P. I. PAVLIK JR, H. CEN, AND K. R. KOEDINGER, *Performance factors analysis–a new alternative to knowledge tracing*, Online Submission, (2009).

[176] D. A. PAYNE, D. R. KRATHWOHL, AND J. GORDON, *The effect of sequence on programmed instruction*, American Educational Research Journal, 4 (1967), pp. 125–132.

[177] J. PIAGET AND P. H. MUSSEN, *Carmichael's manual of child psychology*, 1970.

[178] C. PIECH, J. BASSEN, J. HUANG, S. GANGULI, M. SAHAMI, L. J. GUIBAS, AND J. SOHL-DICKSTEIN, *Deep knowledge tracing*, in Advances in neural information processing systems, 2015, pp. 505–513.

[179] M. POESIO, *The MATE/GNOME proposals for anaphoric annotation, revisited*, in Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue at HLT-NAACL 2004, 2004, pp. 154–162.

[180] S. L. PRESSEY, *A simple apparatus which gives tests and scores-and teaches*, Sch. & Soc., 23 (1926), pp. 373–376.

[181] L. A. RAMSHAW AND M. P. MARCUS, *Text chunking using transformation-based learning*, in Natural language processing using very large corpora, Springer, 1999, pp. 157–176.

[182] M. RANI, A. K. DHAR, AND O. VYAS, *Semi-automatic terminology ontology learning based on topic modeling*, Engineering Applications of Artificial Intelligence, 63 (2017), pp. 108–125.

[183] C. M. REIGELUTH, M. D. MERRILL, AND C. V. BUNDERSON, *The structure of subject matter content and its instructional design implications*, Instructional science, 7 (1978), pp. 107–126.

[184] K. RIES, *Segmenting conversations by topic, initiative, and style*, in Workshop on Information Retrieval Techniques for Speech Applications, Springer, 2001, pp. 51–66.

[185] A. RITTER, S. SODERLAND, AND O. ETZIONI, *What is this, anyway: Automatic hypernym discovery*, in AAAI Spring Symposium: Learning by Reading and Learning to Read, 2009, pp. 88–93.

[186] S. ROLLER, D. KIELA, AND M. NICKEL, *Hearst patterns revisited: Automatic hypernym detection from large text corpora*, arXiv preprint arXiv:1806.03191, (2018).

[187] C. ROMERO AND S. VENTURA, *Educational data mining: a review of the state of the art*, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 40 (2010), pp. 601–618.

[188] D. ROY, S. SARKAR, AND S. GHOSE, *Automatic extraction of pedagogic metadata from learning content*, International Journal of Artificial Intelligence in Education, 18 (2008), pp. 97–118.

[189] S. ROY, M. MADHYASTHA, S. LAWRENCE, AND V. RAJAN, *Inferring concept prerequisite relations from online educational resources*, 31st AAAI Conference on Innovative Applications of Artificial Intelligence (IAAI-19), (2018).

[190] J. C. SAGER, *Practical course in terminology processing*, John Benjamins Publishing, 1990.

[191] J. SALDAÑA, *The coding manual for qualitative researchers*, Sage, 2015.

[192] M. SANDELOWSKI AND J. BARROSO, *Handbook for synthesizing qualitative research*, springer publishing company, 2006.

[193] S. SARAWAGI ET AL., *Information extraction*, Foundations and Trends in Databases, 1 (2008), pp. 261–377.

[194] D. H. SCHUNK, *Learning theories an educational perspective sixth edition*, Pearson, 2012.

[195] E. SCHWARZ, P. BRUSILOVSKY, AND G. WEBER, *World-wide intelligent textbooks*, in EDMEDIA'96-World conference on educational multimedia and hypermedia, 1996.

[196] W. A. SCOTT, *Reliability of content analysis: The case of nominal scale coding*, Public opinion quarterly, (1955), pp. 321–325.

[197] J. SELF, *The defining characteristics of intelligent tutoring systems research: Itss care, precisely*, (1998).

[198] A. SETZER, R. GAIZAUSKAS, AND M. HEPPLE, *Using semantic inference for temporal annotation comparison*, The Language of Time, (2005), p. 575.

[199] M. SHAMSFARD AND A. A. BARFOROUSH, *The state of the art in ontology learning: a framework for comparison*, The Knowledge Engineering Review, 18 (2003), pp. 293–316.

[200] V. SHMELEV, M. KARPOVA, AND A. DUKHANOV, *An approach of learning path sequencing based on revised Bloom's taxonomy and domain ontologies with the use of genetic algorithms*, Procedia Computer Science, 66 (2015), pp. 711–719.

[201] V. SHUTE AND B. TOWLE, *Adaptive e-learning*, Educational psychologist, 38 (2003), pp. 105–114.

[202] V. J. SHUTE, *Rose garden promises of intelligent tutoring systems: Blossom or thorn?*, in NASA Conference Publication, no. 3103, Scientific and Technical Information Office, National Aeronautics and Space, 1991, p. 431.

[203] V. J. SHUTE AND J. PSOTKA, *Intelligent tutoring systems: Past, present, and future*, tech. rep., Armstrong Lab Brooks AFB TX Human Resources Directorate, 1994.

[204] B. F. SKINNER, *Teaching machines*, Science, 128 (1958), pp. 969–977.

[205] R. SNOW, D. JURAFSKY, AND A. Y. NG, *Learning syntactic patterns for automatic hypernym discovery*, in Advances in neural information processing systems, 2005, pp. 1297–1304.

[206] A. SORDONI, Y. BENGIO, AND J.-Y. NIE, *Learning concept embeddings for query expansion by quantum entropy minimization*, in Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014.

[207] S. SOSNOVSKY, P. BRUSILOVSKY, R. AGRAWAL, R. G. BARANIUK, AND A. S. LAN, *Preface*, in First Workshop on Intelligent Textbooks, 2019.

[208] P. STENETORP, S. PYYSALO, G. TOPIĆ, T. OHTA, S. ANANIADOU, AND J. TSUJII, *Brat: a web-based tool for NLP-assisted text annotation*, in Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 2012, pp. 102–107.

[209] M. K. STERN AND B. P. WOOLF, *Curriculum sequencing in a web-based tutor*, in International Conference on Intelligent Tutoring Systems, Springer, 1998, pp. 574–583.

[210] M. STRAKA, J. HAJIC, AND J. STRAKOVÁ, *Udpipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, pos tagging and parsing*, in Proceedings of the tenth international conference on language resources and evaluation (LREC 2016), 2016, pp. 4290–4297.

[211] B. STROUSTRUP, *The C++ programming language*, Pearson Education, 2013.

[212] I. SUBASIC AND B. BERENDT, *From bursty patterns to bursty facts: The effectiveness of temporal text mining for news*, in Proceedings of ECAI 2010: 19th European Conference on Artificial Intelligence, 2010, pp. 517–522.

[213] S. SURESU AND M. ELAMPARITHI, *Probabilistic relational concept extraction in ontology learning*, International Journal of Information Technology, 2 (2016).

[214] P. P. TALUKDAR AND W. W. COHEN, *Crowdsourced comprehension: predicting prerequisite structure in Wikipedia*, in Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, Association for Computational Linguistics, 2012, pp. 307–315.

[215] X. TANNIER AND P. MULLER, *Evaluating temporal graphs built from texts via transitive reduction*, Journal of Artificial Intelligence Research, 40 (2011), pp. 375–413.

[216] R. TESCH, *Qualitative research: Analysis types and software*, Routledge, 2013.

[217] K. THAKER, P. BRUSILOVSKY, AND D. HE, *Student modeling with automatic knowledge component extraction for adaptive textbooks*, in First Workshop on Intelligent Textbooks, 2019.

[218] K. M. THAKER, P. BRUSILOVSKY, AND D. HE, *Concept enhanced content representation for linking educational resources*, in 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), IEEE, 2018, pp. 413–420.

[219] D. THENMOZHI AND C. ARAVINDAN, *An automatic and clause-based approach to learn relations for ontologies*, The Computer Journal, 59 (2016), pp. 889–907.

[220] E. L. THORNDIKE, *Education, a first book*, Macmillan, 1912.

[221] E. L. THORNDIKE, *Educational psychology, vol 2: The psychology of learning*, (1913).

[222] E. L. THORNDIKE AND A. I. GATES, *Elementary principles of education*, (1929).

[223] G. TRENTIN, *Network and mobile technologies in education: a call for e-teachers*, in Using network and mobile technology to bridge formal and informal learning, Elsevier, 2013, pp. 153–182.

[224] J. E. TUOVINEN AND J. SWELLER, *A comparison of cognitive load associated with discovery learning and worked examples*, Journal of educational psychology, 91 (1999), p. 334.

[225] N. UZZAMAN AND J. ALLEN, *Temporal evaluation*, in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, pp. 351–356.

[226] J. VASSILEVA, *Dynamic course generation*, Journal of computing and information technology, 5 (1997), pp. 87–102.

[227] P. VELARDI, S. FARALLI, AND R. NAVIGLI, *Ontolearn reloaded: A graph-based algorithm for taxonomy induction*, Computational Linguistics, 39 (2013), pp. 665–707.

[228] M. VERHAGEN, R. GAIZAUSKAS, F. SCHILDER, M. HEPPLE, G. KATZ, AND J. PUSTEJOVSKY, *SemEval-2007 task 15: Tempeval temporal relation identification*, in Proceedings of the 4th international workshop on semantic evaluations, Association for Computational Linguistics, 2007, pp. 75–80.

[229] J. VILLALON AND R. A. CALVO, *Concept extraction from student essays, towards concept map mining*, in Advanced Learning Technologies, 2009. ICALT 2009. Ninth IEEE International Conference on, IEEE, 2009, pp. 221–225.

[230] J. J. VILLALON AND R. A. CALVO, *Concept map mining: A definition and a framework for its evaluation*, in 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, vol. 3, IEEE, 2008, pp. 357–360.

[231] A. VUONG, T. NIXON, AND B. TOWLE, *A method for finding prerequisites within a curriculum*, in EDM, 2011, pp. 211–216.

[232] L. S. VYGOTSKY, *Mind in society: The development of higher psychological processes*, Harvard university press, 1980.

[233] M. WANG, H. CHAU, K. THAKER, P. BRUSILOVSKY, AND D. HE, *Concept annotation for intelligent textbooks*, arXiv preprint arXiv:2005.11422, (2020).

[234] S. WANG, C. LIANG, Z. WU, K. WILLIAMS, B. PURSEL, B. BRAUTIGAM, S. SAUL, H. WILLIAMS, K. BOWEN, AND C. L. GILES, *Concept hierarchy extraction*

from textbooks, in Proceedings of the 2015 ACM Symposium on Document Engineering, ACM, 2015, pp. 147–156.

[235] S. WANG AND L. LIU, *Prerequisite concept maps extraction for automatic assessment*, in Proceedings of the 25th International Conference Companion on World Wide Web, International World Wide Web Conferences Steering Committee, 2016, pp. 519–521.

[236] S. WANG, A. ORORBIA, Z. WU, K. WILLIAMS, C. LIANG, B. PURSEL, AND C. L. GILES, *Using prerequisites to extract concept maps from textbooks*, in Proceedings of the 25th acm international on conference on information and knowledge management, ACM, 2016, pp. 317–326.

[237] G. WEBER AND P. BRUSILOVSKY, *ELM-ART: An adaptive versatile system for web-based instruction*, (2001).

[238] C. WESTON, T. GANDELL, J. BEAUCHAMP, L. MCALPINE, C. WISEMAN, AND C. BEAUCHAMP, *Analyzing interview data: The development and evolution of a coding system*, Qualitative sociology, 24 (2001), pp. 381–400.

[239] J. E. WISNESKI, G. OZOGUL, AND B. A. BICHELMEYER, *Investigating the impact of learning environments on undergraduate students' academic performance in a prerequisite and post-requisite course sequence*, The Internet and Higher Education, 32 (2017), pp. 1–10.

[240] Y. YANG, H. LIU, J. CARBONELL, AND W. MA, *Concept graph learning from educational data*, in Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, ACM, 2015, pp. 159–168.

[241] C. YIN, N. UOSAKI, H. C. CHU, G.-J. HWANG, J. HWANG, I. HATONO, AND Y. TABATA, *Learning behavioral pattern analysis based on students' logs in reading digital books*, in Proceedings of the 25th international conference on computers in education, 2017, pp. 549–557.

[242] W. C. YOON, S. LEE, AND S. LEE, *Burst analysis of text document for automatic concept map creation*, in International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Springer, 2014, pp. 407–416.

[243] M. V. YUDELSON, K. R. KOEDINGER, AND G. J. GORDON, *Individualized Bayesian knowledge tracing models*, in International conference on artificial intelligence in education, Springer, 2013, pp. 171–180.

[244] J. ZHANG, *The nature of external representations in problem solving*, Cognitive science, 21 (1997), pp. 179–217.

[245] G. ZHAO AND X. ZHANG, *Domain-specific ontology concept extraction and hierarchy extension*, in Proceedings of the 2nd International Conference on Natural Language Processing and Information Retrieval, ACM, 2018, pp. 60–64.

[246] L. ZHOU, *Ontology learning: state of the art and open issues*, Information Technology and Management, 8 (2007), pp. 241–252.

[247] Y. ZHOU AND K. XIAO, *Extracting prerequisite relations among concepts in Wikipedia*, in 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–8.

[248] B. ŽITKO, S. STANKOV, M. ROSIĆ, AND A. GRUBIŠIĆ, *Dynamic test generation over ontology-based knowledge representation in authoring shell*, Expert Systems with Applications, 36 (2009), pp. 8185–8196.

[249] A. ZOUAQ AND R. NKAMBOU, *Building domain ontologies from text for educational purposes*, IEEE Transactions on learning technologies, 1 (2008), pp. 49–62.